



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ESCOLA DE INFORMÁTICA APLICADA

Um estudo sobre Ferramentas de Apoio à Análise de Dados em cenários de
Big Data no Contexto da Computação Cognitiva

Débora França de Oliveira

Orientadoras

Flavia Santoro
Fernanda Baião

RIO DE JANEIRO, RJ – BRASIL
DEZEMBRO DE 2017

Catálogo informatizada pelo autor

O48 Oliveira, Débora França de
Um estudo sobre Ferramentas de Apoio à Análise de
Dados em cenários de Big Data no Contexto da
Computação Cognitiva / Débora França de Oliveira. --
Rio de Janeiro, 2017.
53

Orientadora: Flavia Maria Santoro.
Coorientadora: Fernanda Araujo Baião.
Trabalho de Conclusão de Curso (Graduação) -
Universidade Federal do Estado do Rio de Janeiro,
Graduação em Sistemas de Informação, 2017.

1. Big Data. 2. Computação Cognitiva . 3.
Mineração de dados. I. Santoro, Flavia Maria,
orient. II. Baião, Fernanda Araujo, coorient. III.
Título.

Um estudo sobre Ferramentas de Apoio à Análise de Dados em cenários de
Big Data no Contexto da Computação Cognitiva

Débora França de Oliveira

Projeto de Graduação apresentado à Escola de
Informática Aplicada da Universidade Federal do
Estado do Rio de Janeiro (UNIRIO) para
obtenção do título de Bacharel em Sistemas de
Informação.

Aprovado por:

Flávia Maria Santoro, D.Sc. (UNIRIO)

Fernanda Araujo Baião, D.Sc. (UNIRIO)

Asterio Kiyoshi Tanaka (UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.

DEZEMBRO DE 2017

RESUMO

A quantidade de dados presentes no mundo cresce exponencialmente recebendo grande atenção nos últimos anos. Juntamente com o crescimento dos dados e da tecnologia surgiu a computação cognitiva, um conjunto de técnicas e tecnologias que ainda não possui uma definição universal, mas que possui como uma de suas características o aprendizado, que pode ser realizado a partir da análise de dados. As redes sociais são excelentes fontes de dados uma vez que estas estão disponíveis a todo momento e em qualquer lugar além de refletirem opiniões e sentimentos dos usuários sobre variados assuntos as tornando uma fonte valiosa de enormes quantidades de dados brutos para análises. Essa quantidade e complexidade de dados se encaixa no conceito de *Big Data* que possui como características variedade, volume, veracidade, valor e velocidade. A coleta e mineração de dados é uma solução para analisar essa grande quantidade de dados sendo uma das técnicas que a computação cognitiva abrange. A mineração em tempo real é uma solução que abrange a velocidade, a mais desafiadora das características em cenários de *Big Data*. Uma das plataformas sociais mais utilizadas para mineração de dados é o *Twitter*, que fornece uma API que permite a captura de *tweets* públicos tanto antigos como em tempo real. O presente trabalho apresenta um estudo sobre mineração em tempo real e não real de dados obtidos do *Twitter* em dois momentos distintos utilizando ferramentas com poder de mineração e processamento. Foram utilizadas palavras-chave para limitar a busca de *tweets* à intenção da coleta. Duas ferramentas foram escolhidas para realizar a análise dos dados em dois momentos distintos de coleta minimizando os erros e tempo necessário caso esta fosse realizada de forma manual, além de fornecer novas formas de observar os dados muitas vezes não vislumbradas quando manualmente analisados.

Palavras-chave: *Big Data*, Computação Cognitiva, Mineração de dados, *Twitter*

ABSTRACT

The amount of data present in the world grows exponentially and has been receiving great attention in recent years. Along with the data growth and technology development, cognitive computing has emerged, a set of techniques and technologies that do not yet have a universal definition, but which has as one of its characteristics the learning that can be performed from the data analysis. Social networks are excellent sources of data since they are available anytime, anywhere, and reflect users' opinions and feelings on a variety of subjects, making them a valuable source of massive amounts of raw data for analysis. This amount and complexity of data fits into the concept of Big Data which has as characteristics variety, volume, veracity, value and velocity. Data collection and mining is a solution to analyze this large amount of data being that one of the techniques that cognitive computing encompasses. Real-time data mining is a solution that covers the velocity, the most challenging of the Big Data characteristics. One of the most popular social platforms for data mining is Twitter, which provides an API that allows you to capture both old and real-time public tweets. The present work presents a study on real-time and non-real-time mining of data obtained from Twitter at two different moments using tools with mining power and processing. Keywords were used to limit the search for *tweets* to the intent of the capture. Two tools were chosen to perform the data analysis in two distinct collection moments, minimizing the errors and time required if this was performed manually, as well as providing new insights of the data that are often not seen when manually analyzed.

Keywords: Big Data, Cognitive Computing, Data Mining, Twitter

Índice

1	Introdução	9
1.1	Motivação.....	9
1.2	Objetivos	12
1.3	Organização do texto	12
2	Revisão Bibliográfica.....	13
2.1	Processamento de Dados em Tempo Real.....	13
2.2	Computação Cognitiva	14
2.3	Redes Sociais	17
3	Ferramentas de Suporte à Mineração.....	19
3.1	A suíte ELK (Elasticsearch, Logstash, Kibana)	20
3.2	R.....	Error! Bookmark not defined.
3.3	SparkR	24
4	Metodologia.....	27
4.1	Configuração das Ferramentas.....	28
4.2	Captura e visualização de tweets.....	29
5	Resultados e Discussão.....	33
5.1	Captura de Tweets.....	33
5.2	Utilização do trio ELK	33
5.3	SparkR	39
5.4	Discussões sobre o estudo	45
6	Conclusão	47
	Referências Bibliográficas	50

Índice de Figuras

Figura 1 - Camadas da computação cognitiva. Fonte: TK, S. e Viswanathan, R. [7].....	11
Figura 2 – Como o Watson orienta uma resposta a uma pergunta. Fonte: High, R. [14]	16
Figura 3 - Processo de escolha das ferramentas.....	19
Figura 4 - Arquivo configuração Logstash	21
Figura 5 - Índices no Elasticsearch.....	22
Figura 6 - Escolha do índice no Kibana	22
Figura 7 - Aba de Descoberta do Kibana.....	23
Figura 8 - Aba de Visualização do Kibana.....	23
Figura 9 - Aba de Dashboard do Kibana	24
Figura 10 - Bibliotecas Spark Fonte: Spark [24]	25
Figura 11 - Configuração do SparkR no RStudio	26
Figura 12 - Fluxo de uso das ferramentas.....	28
Figura 13 - Processo de configuração das ferramentas	28
Figura 14 - Inicializando Logstash	29
Figura 15 - Captura de <i>tweets</i> pelo Logstash	30
Figura 16 - Escolha do período para visualização da captura de <i>tweets</i> na aba de descoberta.....	30
Figura 17 -Menu suspenso para criação de gráficos.....	31
Figura 18 - Escolha do período para visualização da captura de <i>tweets</i> para criação de gráficos.....	31
Figura 19 - Primeira configuração arquivo Logstash	34
Figura 20 - Frequência palavras-chave do primeiro dia da primeira coleta no trio ELK.	35
Figura 21 - Dez palavras com maiores frequências ao final da primeira coleta.....	36
Figura 22 - Quantidade de palavras-chave após primeira coleta	36
Figura 23 – Palavras-chave faltantes ou não configuradas no arquivo .conf do Logstash.....	38
Figura 24 - Quantidade de palavras-chave após segunda coleta	38
Figura 25 - Aglomeração palavras-chave capturadas	39

Figura 26 - Contagem de palavras-chave do primeiro dia de coletas.	40
Figura 27 - Modo de aparição das palavras-chave	40
Figura 28 – Diferentes agregações das palavras de acordo com letras maiúsculas e minúsculas.....	40
Figura 29 - Número de ocorrências para a palavra "Impeachment" no trio ELK	41
Figura 30 - Número de ocorrências para a palavra "Impeachment" e ""Impeachment" no SparkR.....	41
Figura 31 - Frequência de palavras-chave precedidas do símbolo jogo da velha no SparkR	42
Figura 32 - Frequência de palavras-chave no SparkR.....	43
Figura 33 - Frequência de palavras-chave na segunda coleta no SparkR	44
Figura 34 - Frequência de palavras-chave na segunda coleta no trio ELK	44

Índice de Tabelas

Tabela 1 - Quadro comparativo das ferramentas Trio ELK e SparkR.....	46
--	----

1 Introdução

1.1 Motivação

Na era da informação pela qual estamos passando, a quantidade de dados disponíveis vem aumentando significativamente [1], e todos esses dados precisam ser armazenados em algum lugar para algum propósito. Além disso, as redes sociais se expandiram nos últimos anos, e isso mudou o modo como as pessoas se comunicam. É notável que um grande número de pessoas gasta uma quantidade significativa de tempo utilizando as redes sociais.

Por outro lado, os dispositivos móveis têm se popularizado muito, especialmente para uso em redes sociais e, aliada à queda nos custos de armazenamento em meios digitais, houve aumento do armazenamento de dados importantes que antes eram descartados. A quantidade de e-mails, mensagens de textos, imagens, áudios e transações produzidos pelas companhias e pessoas físicas é gigantesca e, em grande parte dos casos, se encontra de maneira desordenada.

Acompanhar esse grande fluxo de dados é difícil, porém mais desafiador é analisar a vasta quantidade disponível de dados, especialmente quando não está conforme com a noção tradicional de estrutura de dados, para identificar padrões e extrair informações úteis. Esses desafios relacionados ao dilúvio de dados apresentam uma oportunidade de transformar negócios, governos, ciência e o dia a dia [2].

A promessa de *Big Data* (Conjunto de dados muito grande e complexo [3]) aumentou sobremaneira o potencial de sistemas corporativos de apoio à decisão, devido à possibilidade de uma integração de grandes volumes de dados heterogêneos. Desta forma, conjuntos de dados previamente isolados

podem ser consolidados, analisados e visualizados para apoiar a tomada de decisão organizacional. Este apoio permitirá que os tomadores de decisão decidam mais rápido e com mais confiança em evidências que se manifestam nos repositórios de dados de processos de uma organização.

Big Data possui como características o grande volume de dados em quantidade suficiente para demandar novos paradigmas de armazenamento e recuperação, a complexidade (ou variabilidade) de diversos tipos de dados e estrutura, e a grande velocidade de criação, crescimento e alteração [2]. Desta forma, devido à sua estrutura ou tamanho, não é possível suportar cenários *Big Data* através de bancos de dados ou métodos tradicionais. Seus problemas requerem novas ferramentas e tecnologias para armazenar e gerenciar dados. Essas novas ferramentas e tecnologias devem permitir a criação, manipulação e gerenciamento de grandes conjuntos de dados em diversos formatos e em constante evolução, e do ambiente no qual estão armazenados [2].

Apesar do volume ser a característica que mais se destaca a princípio, a velocidade é a característica mais desafiadora [4]. Essa dimensão, que indiretamente é impactada pela dimensão de volume, diz respeito principalmente à capacidade de lidar com a alta velocidade de chegada dos dados (frequentemente dados em fluxo contínuo, ou *streams*) e a necessidade de alta velocidade do processamento de dados. O desafio mais conhecido que diz respeito a dados em alta velocidade é a transmissão de dados, e a alta velocidade no processamento nesse contexto significa processamento em tempo real [4].

A Computação Cognitiva, área em grande desenvolvimento, ainda não possui uma definição de consenso. No entanto todas as referências mencionam o aprendizado do sistema cognitivo como uma de suas características [5] [6]. A computação cognitiva abrange o processamento de dados como uma de suas tecnologias assim como processamento de linguagem natural.

As tecnologias de computação cognitiva podem tratar de enormes quantidades de dados, aplicar razão, extrair informações e aprender continuamente ao interagir com pessoas e outras máquinas. A Figura 1

demonstra as 3 camadas da computação cognitiva de acordo com a Cognizant [7].

The Blending Layers of Cognitive Computing

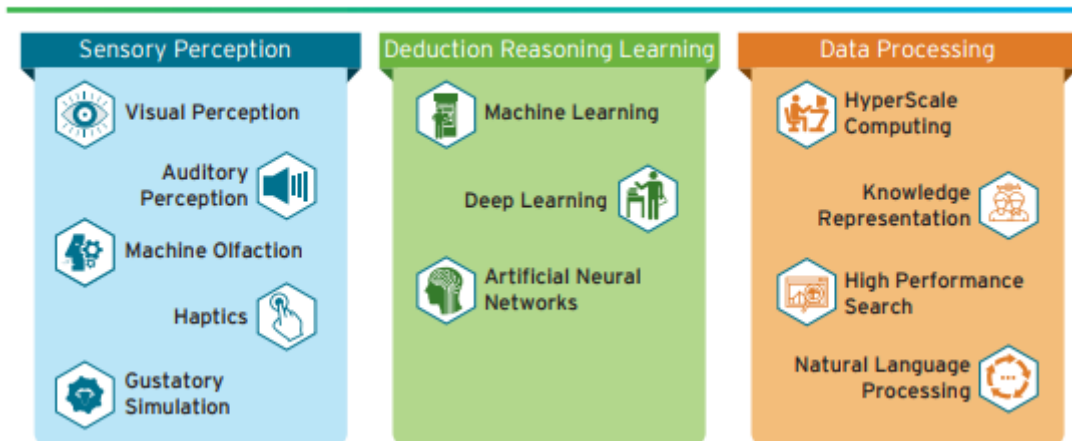


Figura 1 - Camadas da computação cognitiva. Fonte: TK, S. e Viswanathan, R. [7]

A primeira camada descrita pela Cognizant é a camada de percepção sensorial, na qual máquinas são habilitadas para simular sentidos humanos como visão, audição, olfato, toque e paladar. Dentre todos os sentidos, os mais desenvolvidos em termos de simulação de máquina são as percepções visual e auditiva [7]. A camada de dedução, raciocínio e aprendizagem é onde as máquinas simulam o pensamento humano para tomada de decisão. Aprendizado de máquina, aprendizado profundo e redes neurais são proeminentes dentre os paradigmas de tecnologia e já estão sendo implementados como sistemas de inteligência para obter significado a partir de informações e aplicar julgamento [7]. Na camada de processamento de dados, grandes conjuntos de dados são processados para facilitar rápidas decisões comerciais e fornecer sugestões mais inteligentes. Computação de alta escala, representação do conhecimento e ontologias, pesquisa de alto desempenho e processamento de linguagem natural (PLN) são as principais tecnologias nesta camada e fornecem o poder de processamento necessário para garantir que sistemas de engajamento trabalhem em tempo real [7]. Tecnologias de representação do conhecimento, como bancos de dados gráficos, dentre

outras, ajudam os sistemas inteligentes a alcançar ampla e profunda busca de conexões [7].

O presente estudo foi motivado pelo crescimento da computação cognitiva e seus desafios, como a demanda por análise rápida e eficiente de dados que são transmitidos em alta velocidade e quantidade, assim como a descoberta de ferramentas podem ajudar a capturar dados específicos que trafegam em um determinado momento e analisá-los em tempo real.

1.2 Objetivo

O objetivo desse trabalho é avaliar ferramentas de código aberto no âmbito da computação cognitiva, especificamente seu potencial de suporte para tarefas de mineração de grandes quantidades de dados, em tempo real. Desta forma, é proposta a utilização de uma das técnicas de computação cognitiva (processamento de dados em grande escala) para analisar dados não estruturados provenientes do *Twitter*, baseados em uma busca personalizada e usando as ferramentas Trio ELK (composto das ferramentas Elasticsearch, Kibana, Logstash) e SparkR e são feitas comparações entre resultados obtidos.

1.3 Organização do texto

O presente trabalho está estruturado em capítulos, além desta introdução.

O Capítulo 2 abrange a revisão bibliográfica, trazendo conceitos chave estudados neste trabalho e das ferramentas utilizadas para sua realização. O Capítulo 3 descreve as ferramentas utilizadas no estudo. O Capítulo 4 descreve a metodologia adotada para análise das ferramentas. O Capítulo 5 compreende os resultados e discussão, onde são descritos os resultados no trabalho desde a captura de *tweets* até a visualização dos dados obtidos. O Capítulo 6 abrange as considerações finais, assinalando as contribuições da pesquisa e os possíveis trabalhos futuros.

2 Revisão Bibliográfica

2.1 Processamento de Dados em Tempo Real

Com o avanço tecnológico e o aumento da conectividade entre pessoas e dispositivos, a quantidade de dados disponíveis aumenta exponencialmente. O surgimento da web 2.0 e da Internet das coisas tornou viável rastrear diversos tipos de informações ao longo do tempo, em particular as atividades de fina granularidade dos usuários e dados de sensores em seu ambiente e até sua biometria. No entanto, enquanto a eficiência continua obrigatória para qualquer aplicação tentando lidar com grandes quantidades de dados, apenas parte do potencial dos grandes repositórios de dados de hoje pode ser explorado usando a abordagem tradicional *em batch*, uma vez que o valor dos dados geralmente se deteriora rapidamente e alta latência torna-se inaceitável em algumas aplicações [8].

As condições dinâmicas de ambiente, menor ciclo de vida dos produtos e aumento da complexidade no ambiente da manufatura são apenas alguns problemas que as empresas enfrentam hoje. Para lidar com esses problemas as empresas precisam adaptar seus processos [9].

A mudança para conteúdo mais dinâmico e gerado por usuários na web e a onipresença de telefones inteligentes, dispositivos vestíveis e outros dispositivos móveis em particular levaram a uma abundância de informações que são apenas valiosas por um curto período de tempo e, portanto, devem ser processadas imediatamente [8].

A dimensão de volume relacionada aos dados, apesar de ser um problema antigo, está sendo escalonada atualmente na busca de novos paradigmas de recuperação e processamento de dados. A dimensão de variedade é focada em dados semiestruturados e não estruturados em novas

aplicações. A dimensão de velocidade consiste tanto em alta velocidade da chegada de dados quanto na necessidade da alta velocidade no processamento de dados [4].

Transmissão em cenários de *Big Data* possui algumas características marcantes. Nestes contextos, os dados são recebidos continuamente, infinitamente, explosivamente, imprevisivelmente e numa sequência variável no tempo [4]. Portanto, processamento de *Big Data* em tempo real permite que os tomadores de decisão decidam mais rápido e com mais confiança na evidência real daquele momento do que a que se manifesta nos repositórios de dados, não tão recentes, de processos de uma organização.

2.2 Computação Cognitiva

O cérebro humano é um modelo ideal [10]. É pequeno, eficiente e pode processar diversos tipos de estímulos instantaneamente. Computadores são comparados ao cérebro humano desde o início do estabelecimento da ciência da computação. No entanto, computadores ainda não implementam diversas funções cognitivas que são fáceis para humanos. Com algum treinamento, computadores são capazes de reconhecer e rotular com acurácia um gato e um banco, por exemplo. Mas possuem dificuldade em narrar um vídeo de um gato pulando sobre um banco [10].

Sistemas cognitivos têm como objetivo imitar atividades humanas como percepção, dedução, reunir evidências, desenvolver hipóteses e raciocinar. Quando combinados com automação avançada, esses sistemas podem ser treinados para executar tarefas que necessitam julgamento intensivo [11]. Sistemas cognitivos tentam imitar os aspectos do pensamento humano adicionando a habilidade de lidar com grandes quantidades de informações e avaliá-las imparcialmente [12].

A área da computação cognitiva ainda está emergindo e ainda não há uma definição amplamente definida. Cada instituição possui seu diferente ponto de vista sobre o que computação cognitiva é e como afetará o mundo [13]. De acordo com a IBM [5], computação cognitiva se refere a sistemas que

aprendem em escala, racionalizam objetivamente e interagem com humanos naturalmente. Em vez de serem explicitamente programados, eles aprendem e racionalizam a partir de suas interações conosco e de suas experiências com seu ambiente.

A Deloitte [6] afirma que diferentemente dos sistemas computacionais tradicionais que são programados por pessoas para desenvolverem certas tarefas, sistemas cognitivos podem aprender a partir de experiência e instrução. O poder da computação cognitiva é sua habilidade de digerir, tanto dados estruturados, quanto não estruturados e chegar a conclusões a partir deles, simulando o cérebro humano e realizando tarefas que tradicionalmente apenas pessoas eram capazes de realizar. Sistemas cognitivos podem processar informações além da capacidade humana, identificando padrões e provendo soluções em potencial que humanos talvez nunca reconhecessem através de análises tradicionais. A Deloitte afirma que “Computação cognitiva detém o potencial para remodelar o modo de trabalho, crescimento dos negócios e como mercados e indústrias evoluem.”

Sistemas cognitivos são probabilísticos [5]. São desenhados para se adaptarem e terem senso de complexidade e imprevisibilidade de informações não estruturadas. Eles podem “ler” textos, “ver” imagens e “ouvir” fala natural. Podem interpretar esta informação, organizá-la e oferecer explicações do que significam juntamente com o raciocínio para suas conclusões. Eles não oferecem respostas definitivas pois na verdade não “sabem” a resposta. São desenhados para verificar as informações e ideias de múltiplas fontes, raciocinar e depois oferecer a hipótese para consideração. Um sistema cognitivo atrela um nível de confiança a cada potencial resposta [5].

Na Figura 2 pode-se ver como o Watson¹, sistema cognitivo desenvolvido pela IBM, orienta uma resposta a uma pergunta. O Watson identifica a pergunta e respostas potenciais no corpus (todos os tipos de conhecimento não estruturado, como livros de texto, diretrizes, manuais de instruções, perguntas frequentes, planos de benefícios e notícias.), compara profundamente a pergunta e seu contexto com cada potencial resposta em

¹ <https://www.ibm.com/watson>

centenas de maneiras, e então usa os resultados para obter um grau de confiança em sua interpretação da pergunta e de potenciais respostas [14].

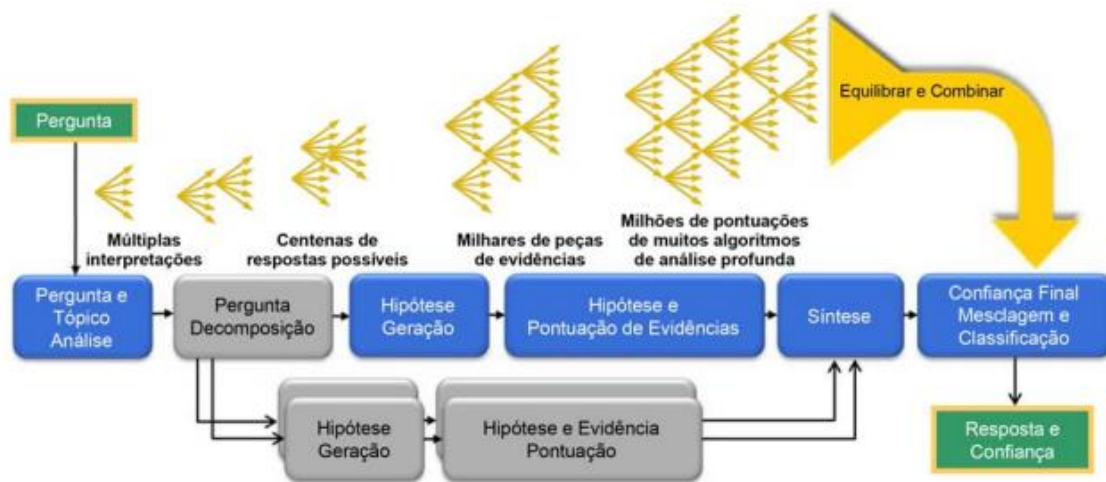


Figura 2 – Como o Watson orienta uma resposta a uma pergunta. Fonte: High, R. [14]

Dentro da comunidade computacional, a definição de computação cognitiva é um tópico de debate. É geralmente associado com inteligência artificial (IA), um campo da tecnologia que abrange aspectos gerais da inteligência humana. IA inclui as habilidades relacionadas ao raciocínio e resolução de problemas, mas também a percepção (reconhecimento e visão facial) e a capacidade de manipular objetos (robótica) [12].

Alguns aspectos da computação cognitiva, como a capacidade de abordar o volume, a velocidade, a variedade e a veracidade dos dados, não são áreas de foco na comunidade de desenvolvimento da IA. As tecnologias cognitivas são necessárias porque abordam desafios de dados aplicando múltiplas tecnologias para permitir a compreensão de fontes de dados vastas e díspares em uma única solução. Através de uma abordagem abrangente para a agregação, compreensão e análise de dados, juntamente com tecnologias que leem, fundamentam e aprendem, podem ser descobertas novas vias de pesquisa [12] como por exemplo, o modelo de processo de negócios cognitivo descritos em [13, 15] onde a computação cognitiva terá papel fundamental pois esta pode acelerar a chegada da nova geração de Gestão de Processos de Negócios [13].

Assim, a computação cognitiva abrange uma família de tecnologias emergentes que inclui processamento de linguagem natural (PLN), aprendizado de máquinas, mineração de dados e a habilidade de melhoria do sistema através de aprendizado experimental irá possibilitar aquisição de conhecimento em escala. De acordo com [13], os serviços presentes no escopo da computação cognitiva serão amplamente disponíveis e relativamente baratos nos próximos anos, permitindo a computação cognitiva constante. O sucesso da computação cognitiva não será medido por testes de Turing ou pela capacidade de um computador imitar humanos. Será mensurada em meios mais práticos como retorno de investimentos, novas oportunidades de mercado, cura de doenças e vidas salvas [5].

A computação cognitiva tem potencial para alavancar e melhorar a colaboração entre humanos devido à capacidade de ingerir e raciocinar comunicações em linguagem natural e irá melhorar a colaboração máquina-humano através de melhor comunicação e das máquinas serem muito melhores em entender e cumprir objetivos e intenções humanas [13].

2.3 Redes Sociais

A noção de uma rede social e os métodos de análise de rede social atraíram interesse e curiosidade consideráveis da comunidade e da ciência comportamental nas últimas décadas. Muito desse interesse pode ser atribuído ao foco atraente da análise de rede social em relacionamentos entre entidades sociais e sobre os padrões e implicações desses relacionamentos [16]. *Facebook*², *Twitter*³, e *LinkedIn*⁴ são exemplos de domínios de aplicação em redes sociais. Isso se deve ao crescimento dos sites de relacionamentos, atingindo um grande público e impactando drasticamente o modo de comunicação da população.

Os sites de relacionamentos mais acessados são *Facebook*, *Twitter* e *Instagram*⁵ ocupando a terceira, décima terceira e décima oitava posições

² <https://www.facebook.com>

³ <https://www.twitter.com>

⁴ <https://www.linkedin.com>

⁵ <https://www.instagram.com/>

respectivamente⁶. Diversas pesquisas têm sido voltadas para mineração de dados de redes sociais para detectar eventos e notícias [17]. A análise dos dados coletados também aumenta o entendimento sobre os consumidores que são importantes recursos para empresas que lidam com *Big Data*. Dados também podem ser extraídos usando aplicações tais como monitoramento do fluxo.

Redes sociais que, tais como o *Twitter*, compartilham mensagens curtas, se tornaram ferramentas poderosas e baratas para extração de diversos tipos de informação. O *Twitter* é muito utilizado, e uma significativa parte de seus dados que são compartilhados por indivíduos para o público podem ser adquiridos utilizando-se *Application Programming Interfaces* (APIs). O *Twitter* produz mais de 340 milhões de *tweets* (mensagens curtas ou *posts*) por dia por mais de 140 milhões de usuários [17].

O *Twitter* oferece dois tipos de APIs de rastreamento que permitem aos usuários consultarem os *tweets* por palavras-chave, IDs de usuários e hora / data. O primeiro tipo, REST API, permite aos usuários enviarem consultas para recuperar *tweets* não sendo tempo real. O segundo tipo, Streaming API, possibilita a captura dos *tweets* em tempo real.

Ambas as APIs se conectam com diversas linguagens e programas como Java, .NET, Python, Elasticsearch. No entanto, para utilizar essas APIs é necessário criar uma aplicação utilizando uma conta do *Twitter* e gerar os códigos de acesso, sendo que dois deles são secretos e únicos para cada usuário. O *Twitter* usa *OAuth* para fornecer acesso autorizado à sua API. *OAuth* é um protocolo aberto para permitir a autorização segura em um método simples e padrão de aplicações web, móveis e desktop [18].

No capítulo seguinte, são apresentadas as ferramentas escolhidas para a realização deste estudo.

⁶ <https://www.alexacomtopsites>

3 Ferramentas de Suporte à Mineração

O presente capítulo objetiva apresentar as ferramentas estudadas, descrevendo suas principais características, interfaces e como foram utilizadas na pesquisa. Todas as ferramentas utilizadas são de código aberto e seu processo de escolha pode ser visualizado na Figura 3.

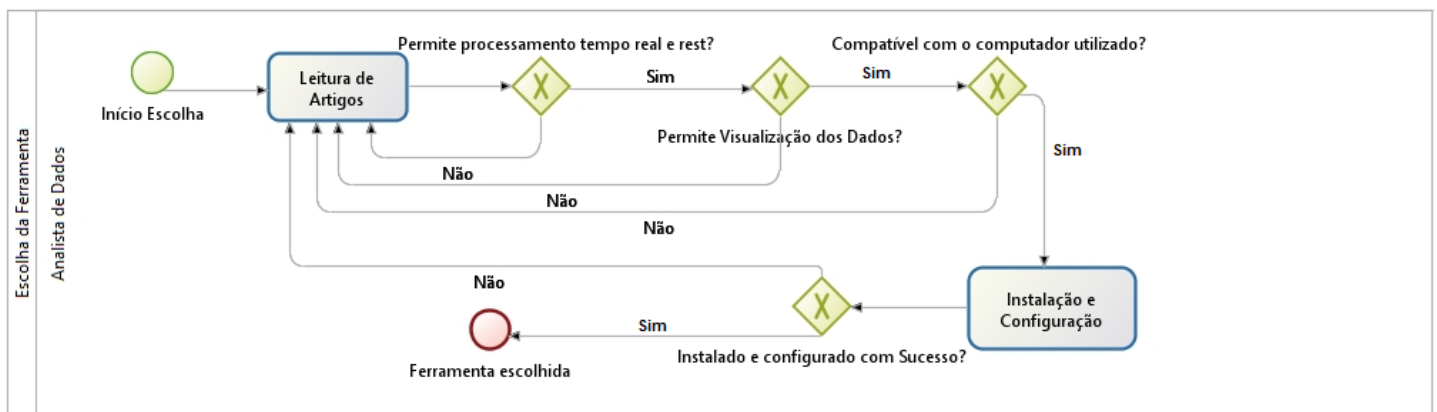


Figura 3 - Processo de escolha das ferramentas

O termo "código aberto" refere-se a algo que as pessoas podem modificar e compartilhar porque seu código-fonte é acessível ao público [19]. O termo se originou no contexto do desenvolvimento de software para designar uma abordagem específica para a criação de programas de computador. Hoje, no entanto, "código aberto" designa um conjunto mais amplo de valores - o que chamamos de "caminho aberto". Os projetos, produtos ou iniciativas de código aberto adotam e celebram princípios de intercâmbio aberto, participação colaborativa, prototipagem rápida, transparência, meritocracia e desenvolvimento orientado para a comunidade [19].

O software livre, de acordo com [20], é uma questão de liberdade dos usuários para executar, copiar, distribuir, estudar, alterar e melhorar o software.

Um programa é um software livre se os usuários tiverem todas essas liberdades. Assim, você deve ser livre para redistribuir cópias, com ou sem modificações, gratuitamente ou cobrando uma taxa para distribuição, a qualquer pessoa em qualquer lugar [20]. Em geral, as licenças de código aberto concedem aos usuários do computador permissão para usar o software de código aberto para qualquer finalidade que desejem [19].

3.1 A suíte ELK (Elasticsearch, Logstash, Kibana)

O Elasticsearch, Logstash e Kibana (ELK) são 3 ferramentas, fornecidas pela Elastic, que compõem uma arquitetura de suporte a análise, captura, indexação e visualização de dados. O Logstash captura os dados, os indexa no Elasticsearch e estes são visualizados no Kibana. Por este motivo serão tratadas em conjunto nessa seção.

O Elasticsearch é uma plataforma altamente escalável, de código aberto, para a busca, indexação e análise de dados. Permite armazenar, buscar e analisar grande volume de dados rapidamente, praticamente em tempo real. É um servidor de dados que suporta vários formatos de linguagens de buscas. Geralmente utilizada como base para outras tecnologias que possuem buscas complexas [21].

O Logstash também possui código aberto, e implementa um *pipeline* que ajuda a processar registros históricos (logs) de dados, além de ser horizontalmente escalável (adicionar mais nós a um sistema (como máquinas virtuais) ou removê-los, se necessário [22]). Ele pode dinamicamente unificar dados de fontes diferentes e normalizá-los no destino de sua preferência. Esta aplicação recebe logs de diversas fontes, sendo o *Twitter* uma delas, e indexa essas informações no Elasticsearch [21].

Kibana é uma plataforma analítica e visual desenhada para trabalhar com o Elasticsearch. Kibana é utilizado para buscar, ver e interagir com os dados armazenados nos índices do Elasticsearch. Facilmente se obtém análise

visual dos dados via gráficos, tabelas e mapas. A análise e visualização dos dados podem ser feitas em tempo real. [21].

No presente estudo, o Logstash irá receber dados do *Twitter* e os indexa no Elasticsearch, fazendo a conexão entre o Elasticsearch e o Kibana e permitindo a visualização dos *tweets* capturados pelo Elasticsearch.

O Logstash necessita de um arquivo de configuração (Figura 4) para coleta dos *tweets*, onde é necessário colocar os códigos de aplicação do *Twitter consumer key* e *consumer secret*. Os campos *Oauth* são utilizados para autorização segura. O campo *Keywords* é um array de todas as palavras que serão procuradas no *tweet*. Estas devem ser descritas separadas por vírgulas e entre aspas duplas.

Por último, o *output* define o tipo de configuração que será usado para compactação e descompactação dos dados que serão enviados ao Kibana e o *índice* é a identificação desta configuração que pode ser visualizada no Elasticsearch (Figura 5) e também será utilizada no Kibana.

```
input {
  twitter{
    # add your data
    consumer_key => " "
    consumer_secret => " "
    oauth_token => " "
    oauth_token_secret => " "
    keywords => ["#LulaInocente", "#ForaTemer", "#impeachment", "#Impeachment",
    "#foratemer", "#lulainocente", "#LavaJato", "#lavajato", "#LulaNaCadeia",
    "#lulanacadeia", "#Moro", "#Lula2018", "#lavajato"]
  }
}
filter { }
output {
  stdout{codec => dots}
  elasticsearch{
    hosts => "localhost:9200"
    index => "lavajato"
  }
}
```

Figura 4 - Arquivo configuração Logstash

```
c:\IC\elasticsearch>curl "localhost:9200/_cat/indices?v"
health status index      pri rep docs.count docs.deleted store.size pri.store.size
yellow open   dilma        5    1   360224         0    193.4mb    193.4mb
yellow open   twitter      5    1     6436         0      4.4mb     4.4mb
yellow open   .kibana       1    1       28         0      35.7kb     35.7kb
yellow open   lavajato     5    1    59229         0     70.7mb     70.7mb
```

Figura 5 - Índices no Elasticsearch

Ao iniciar o Kibana no navegador, um índice do Elasticsearch terá que ser escolhido, como pode ser visto na Figura 6. O índice do estudo foi criado através do arquivo de configuração do Logstash.

Uma vez que o índice é configurado no Kibana e os *tweets* começam a ser coletados, pode-se visualizar a quantidade coletada, de acordo com o *timestamp* desejado na aba de Descoberta. Na aba de Visualização, criam-se gráficos, tabelas, mapas entre outros.

Configure an index pattern

In order to use Kibana you must configure at least one index pattern. Index patterns are used to identify the Elasticsearch index to run search and analytics against. They are also used to configure fields.

Index pattern [advanced options](#)

⚠ Unable to fetch mapping. Do you have indices matching the pattern?

Patterns allow you to define dynamic index names using `*` as a wildcard. Example: `logstash-*`

Time Filter field name [refresh fields](#)

☐ Expand index pattern when searching [DEPRECATED]

With this option selected, searches against any time-based index pattern that contains a wildcard will automatically be expanded to query only the indices that contain data within the currently selected time range.

Searching against the index pattern `logstash-*` will actually query Elasticsearch for the specific matching indices (e.g. `logstash-2015.12.21`) that fall within the current time range.

With recent changes to Elasticsearch, this option should no longer be necessary and will likely be removed in future versions of Kibana.

☐ Use event times to create index names [DEPRECATED]

Time Filter field name is required

Figura 6 - Escolha do índice no Kibana

O Kibana também possibilita a criação de Dashboards com os gráficos criados, na aba de Visualização. Um dashboard pode conter diversos tipos de

visualizações. As Figuras 7, 8 e 9 mostram as interfaces de Descoberta, Visualização e Dashboard do Kibana.

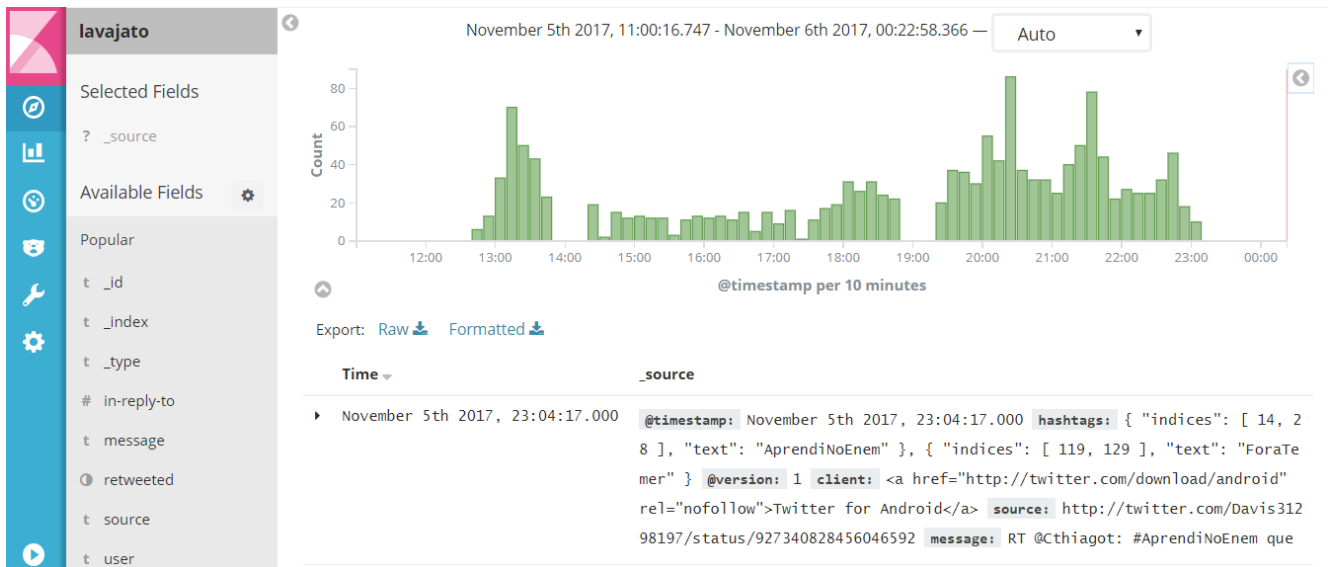


Figura 7 - Aba de Descoberta do Kibana

Select visualization type

Search visualization types...

Basic Charts



Data



Maps

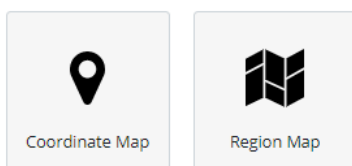


Figura 8 - Aba de Visualização do Kibana

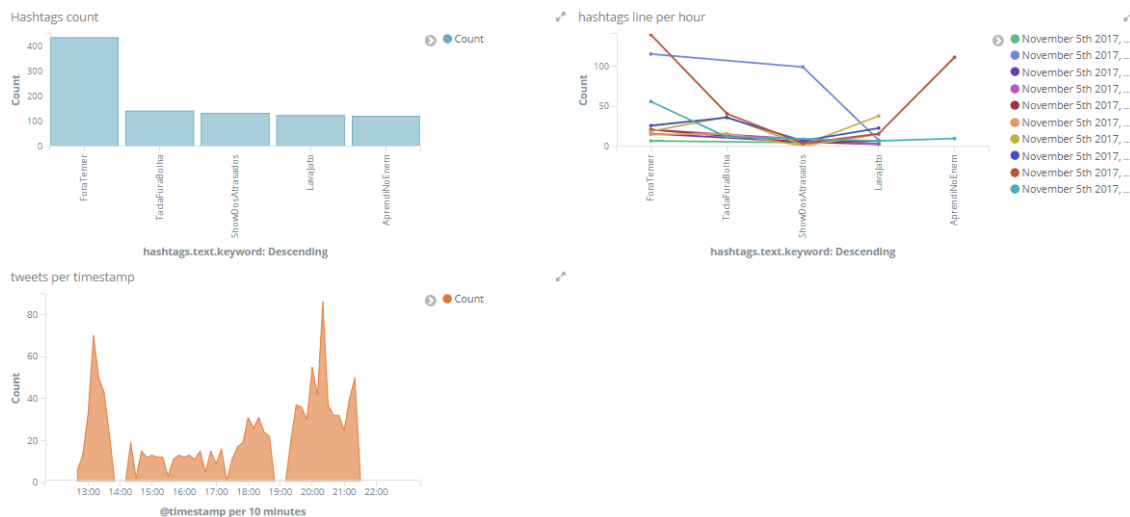


Figura 9 - Aba de Dashboard do Kibana

3.2 SparkR

O Apache Spark é um sistema de computação de cluster rápido e de propósito geral. Ele fornece APIs de alto nível em Java, Scala, Python e R e um mecanismo otimizado que suporta gráficos de execução geral. É um framework para processamento de *Big Data* construído com foco em velocidade, facilidade de uso e análises sofisticadas. Permite realizar análises rápidas em quantidades massivas de dados, baseados em estruturas de memória compartilhadas e integrado ao ambiente *Hadoop* (estrutura que permite o processamento distribuído de grandes conjuntos de dados em *clusters* de computadores usando modelos de programação simples [23]). Ele também suporta um conjunto rico de ferramentas, incluindo Spark SQL para SQL e processamento de dados estruturados, MLlib para aprendizagem em máquina, GraphX para processamento de gráficos e Spark Streaming (Figura 10) [24].

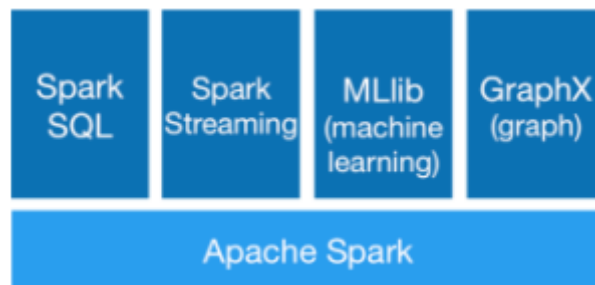


Figura 10 - Bibliotecas Spark Fonte: Spark [24]

O SparkR é um pacote R que fornece uma interface leve para usar o Apache Spark a partir do R. O SparkR suporta operações como seleção, filtragem, agregação em grandes conjuntos de dados [25].

R é uma linguagem e ambiente para computação estatística e gráficos. R fornece uma grande variedade de funções estatísticas (modelagem linear e não-linear, testes estatísticos clássicos, análise de séries temporais, classificação, agrupamento, dentre outros) e técnicas gráficas, além de ser altamente extensível [26]. Um dos pontos fortes de R é a facilidade com que se pode produzir gráficos de alta qualidade, incluindo símbolos matemáticos e fórmulas, quando necessário [26]. O núcleo de R é uma linguagem interpretada que permite ramificação e loop, bem como programação modular usando funções [27].

O R, no ambiente RStudio, foi utilizado em conjunto com o Spark facilitando a manipulação dos dados, uma vez que este promove de forma mais simplificada a análise dos dados e gráficos. Em união com o Spark, a quantidade de dados suportada para análise pelo R aumenta de modo que é possível análise de grandes quantidades de dados.

Durante o presente estudo foi utilizado o SparkR para analisar a grande quantidade de dados obtida através da configuração do Elasticsearch e Logstash.

Após inicializar o SparkR na linha de comando, há a necessidade de se iniciar a sessão do SparkR no Rstudio através do comando `sparkR.session()`. Foi necessário também de instalar o pacote Sparklyr e chamar a biblioteca SparkR além de definir o caminho do SparkR no computador pelo comando `Sys.setenv(SPARK_HOME = " ")`, como pode ser visto na Figura 11.

```
> install.packages(Sparklyr)
> Sys.setenv(SPARK_HOME = "C:/sparkR")
> .libPaths(c(file.path(Sys.getenv("SPARK_HOME"), "R", "lib"), .libPaths()))
> library(SparkR)
> sparkR.session(master = "local[*]", sparkConfig = list(spark.driver.memory = "2g"))
```

Figura 11 - Configuração do SparkR no RStudio

Uma vez que os dados foram coletados apenas pela ferramenta Logstash, estes foram exportados para um arquivo .csv. Após a configuração inicial do SparkR no RStudio, o arquivo .csv é atribuído a um RDD (*Resilient Distributed Dataset*). RDD é uma coleção de elementos tolerantes a falhas que podem ser operados em paralelo. Existem duas maneiras de criar RDDs: paralelizar uma coleção existente em seu programa de driver ou fazer referência a um conjunto de dados em um sistema de armazenamento externo [28]. A partir do RDD que contém o arquivo .csv, as palavras-chave são extraídas e contadas.

4 Metodologia

Este capítulo visa apresentar cenários de uso das ferramentas escolhidas para análise de grande quantidade de dados coletados do *Twitter* assim como sua configuração. Os testes visam avaliar suas funcionalidades frente uma das tecnologias que compõe a computação cognitiva, o processamento de grandes quantidades de dados, em tempo real e não real.

A base de dados foi extraída do *Twitter*, e a coleta foi feita em dois momentos distintos, ambos em tempo real, com palavras-chave diferentes: “foradilma”, “foracunha”, “impeachment”, “naovaitergolpe” para o primeiro momento e “#LulaInocente”, “#ForaTemer”, “#impeachment”, “#Impeachment”, “#foratemer”, “#lulainocente”, “#LavaJato”, “#lavajato”, “#LulaNaCadeia”, “#lulanacadeia”, “#Moro”, “#Lula2018”, para o segundo momento.

A Figura 12 apresenta o fluxo do uso das ferramentas. O arquivo de configuração com as palavras-chave alimenta o Logstash que cria um novo índice no Elasticsearch. Este, por sua vez, envia as informações do índice ao Kibana. Os *tweets* que chegaram ao Kibana foram exportados para um arquivo CSV e importados no RStudio utilizando o SparkR.

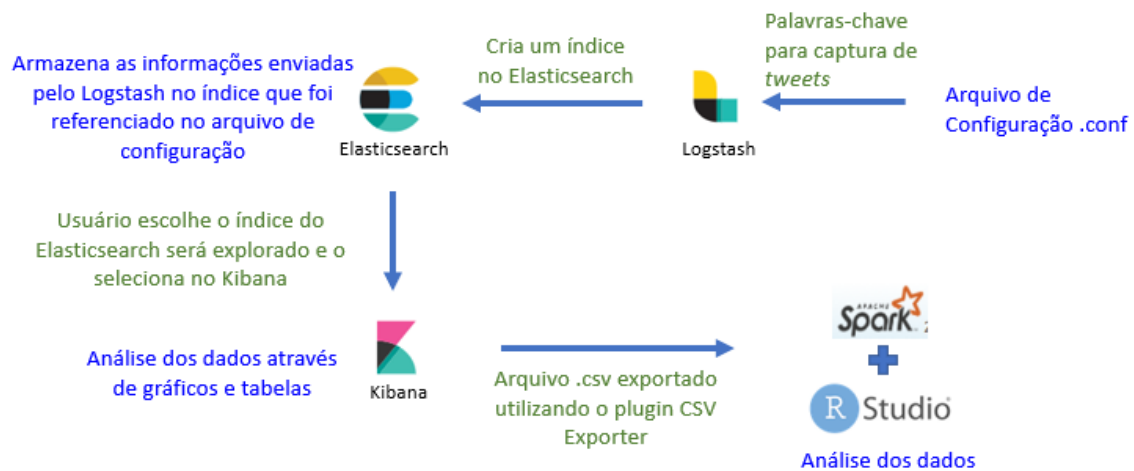


Figura 12 - Fluxo de uso das ferramentas

4.1 Configuração das Ferramentas

A Figura 13 representa o processo de configuração das ferramentas utilizadas. O trabalho se iniciou com a escolha das ferramentas após estudo das ferramentas de código aberto para mineração em tempo real mais utilizadas. Após a escolha, foi necessário configurar todas.

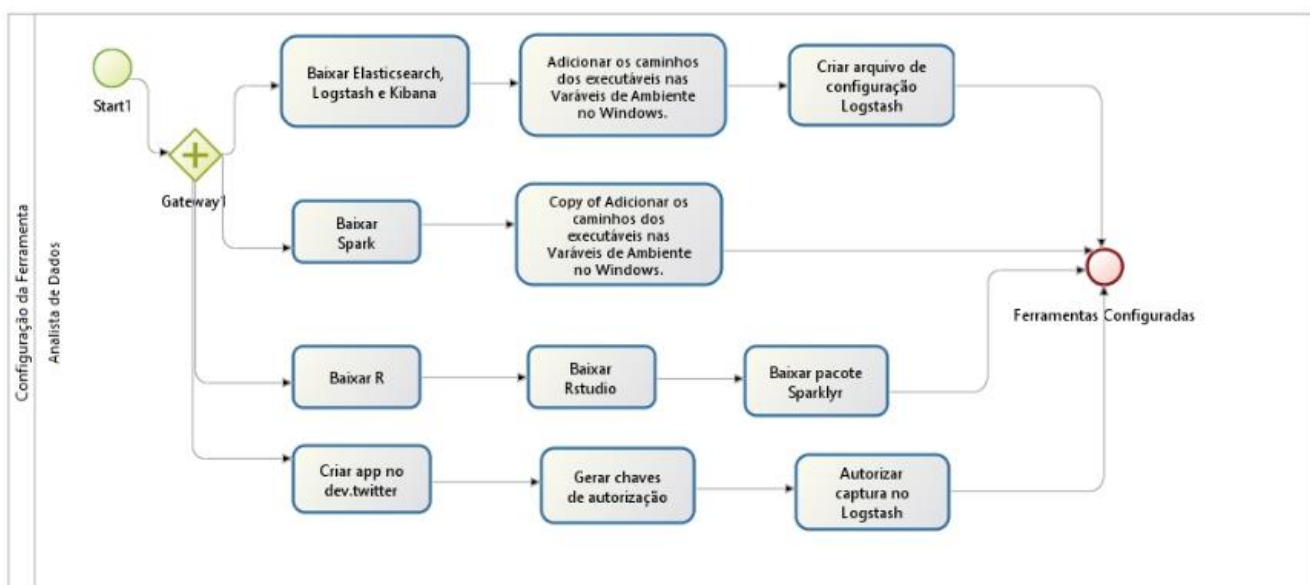


Figura 13 - Processo de configuração das ferramentas

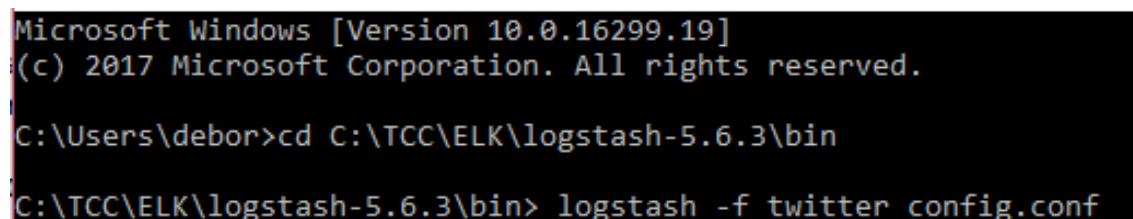
O trio ELK, composto pelas ferramentas Elasticsearch, Logstash e Kibana foi baixado e configurado. É necessário adicionar todas ferramentas nas variáveis de ambiente do sistema operacional. O segundo passo é criar a conexão entre Elasticsearch e Logstash via arquivo de configuração (.conf). Este arquivo, como supramencionado, contém chaves de autenticação da aplicação do *Twitter*, as palavras-chave que serão buscadas, além do host e índice que será criado no Elasticsearch e para onde os *tweets* serão direcionados. O índice criado no Elasticsearch será utilizado para configurar o Kibana, onde é possível visualizar todos os dados que chegam do Logstash.

O SparkR é uma extensão do Spark e está disponível dentro do arquivo Spark que é baixado. Assim como o trio ELK, tanto o Spark quanto o SparkR também devem ser configurados nas variáveis de ambiente. É necessário possuir o R instalado para que o SparkR seja inicializado na linha de comando.

Foi escolhido o RStudio como interface para o ambiente R, que irá servir de plataforma para o SparkR. Sua configuração consiste em instalar o pacote Sparklyr e chamar a biblioteca SparkR.

4.2 Captura e visualização de tweets

A captura de *tweets* se deu apenas pelo trio ELK. Para tal, é necessário inicializar as três ferramentas na linha de comando, sendo que o Logstash deve ser inicializado com o arquivo de configuração atrelado pelo comando “logstash -f” seguido do arquivo de configuração, como pode ser visualizado na Figura 14.



```
Microsoft Windows [Version 10.0.16299.19]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\debor>cd C:\TCC\ELK\logstash-5.6.3\bin
C:\TCC\ELK\logstash-5.6.3\bin> logstash -f twitter_config.conf
```

Figura 14 - Inicializando Logstash

Uma vez inicializado, o Logstash escreve um ponto para cada *tweet* coletado como pode-se verificar na Figura 15. O Logstash também exibe as palavras-chave utilizadas no arquivo de configuração e qual a porta do hospedeiro.

```
[2017-11-08T23:29:58,024][INFO ][logstash.pipeline] Pipeline main started
[2017-11-08T23:29:58,026][INFO ][logstash.inputs.twitter] Starting twitter tracking
{:track=>"#ForaTemer,#foratemer,#LavaJato,lavajato"}
[2017-11-08T23:29:58,293][INFO ][logstash.agent] Successfully started Logst
ash API endpoint {:port=>9600}
.....
```

Figura 15 - Captura de *tweets* pelo Logstash

Uma vez que o Elasticsearch e o Logstash foram inicializados, inicializa-se o Kibana. Este possui uma interface web que é possível acessar através do endereço default localhost:5601. Foi criada uma aba de descoberta para o índice utilizado no arquivo de configuração do Logstash. Nesta descoberta foi possível acompanhar a captura dos *tweets* em tempo real, assim como a captura em momentos específicos como pode ser visto na Figura 16.

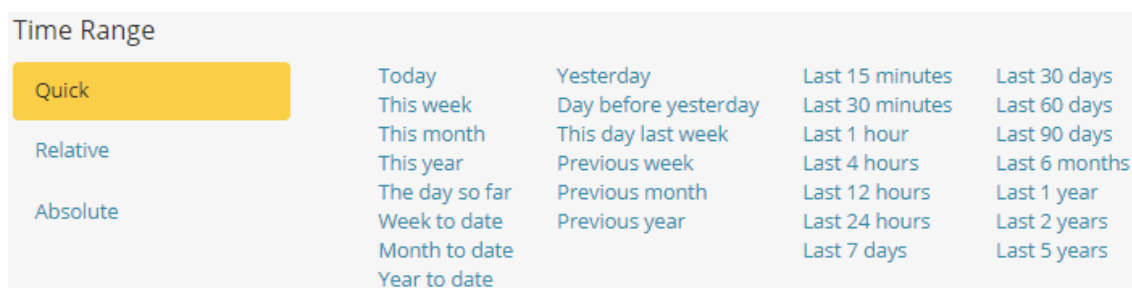


Figura 16 - Escolha do período para visualização da captura de *tweets* na aba de descoberta

Após verificar o início da coleta e que os *tweets* estavam sendo importados para o Kibana corretamente, foram criados gráficos na aba de visualização para melhor acompanhar as mudanças à medida que aconteciam. A ferramenta proporciona a criação de diversos tipos de gráficos com facilidade pois apresenta interface intuitiva e opções tanto de agrupamento quanto de campos e ordenação (Figura 17). Da mesma forma que é possível escolher o momento que deseja visualizar os *tweets* na aba de descoberta, também há essa possibilidade na aba de visualização (Figura 18).

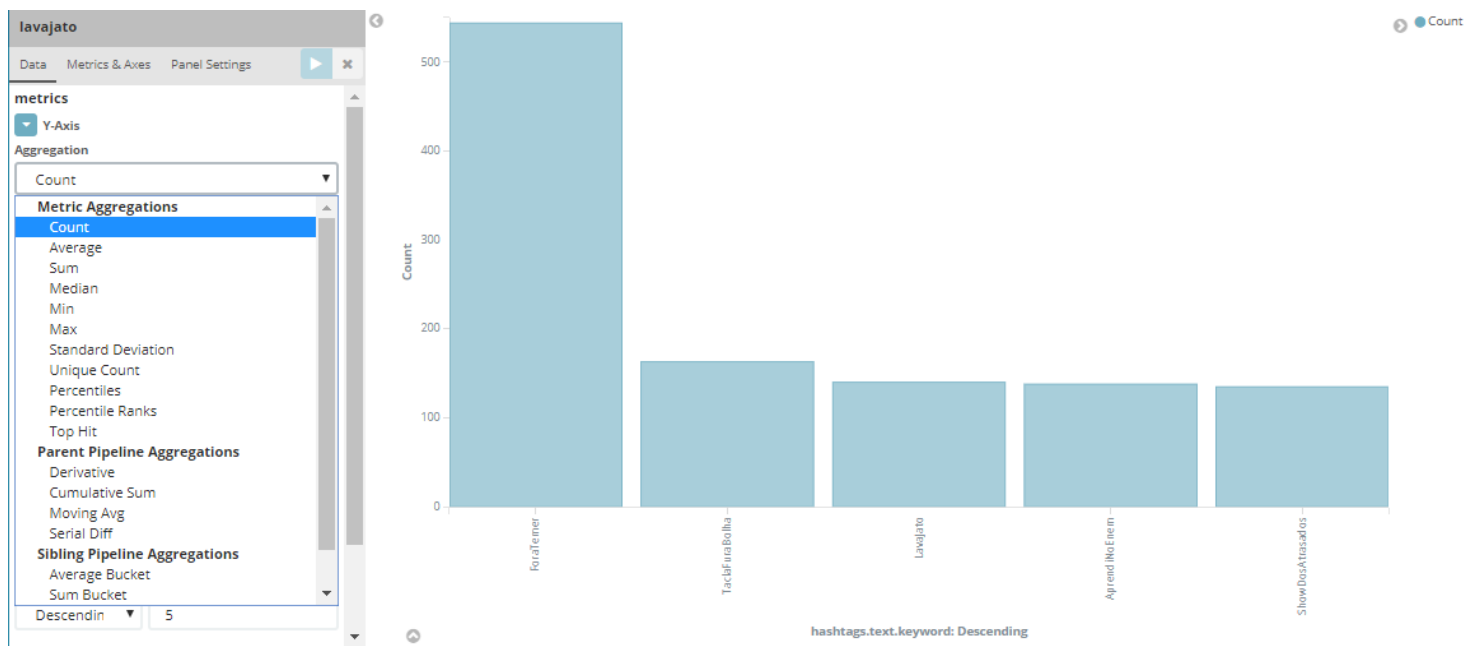


Figura 17 -Menu suspenso para criação de gráficos.

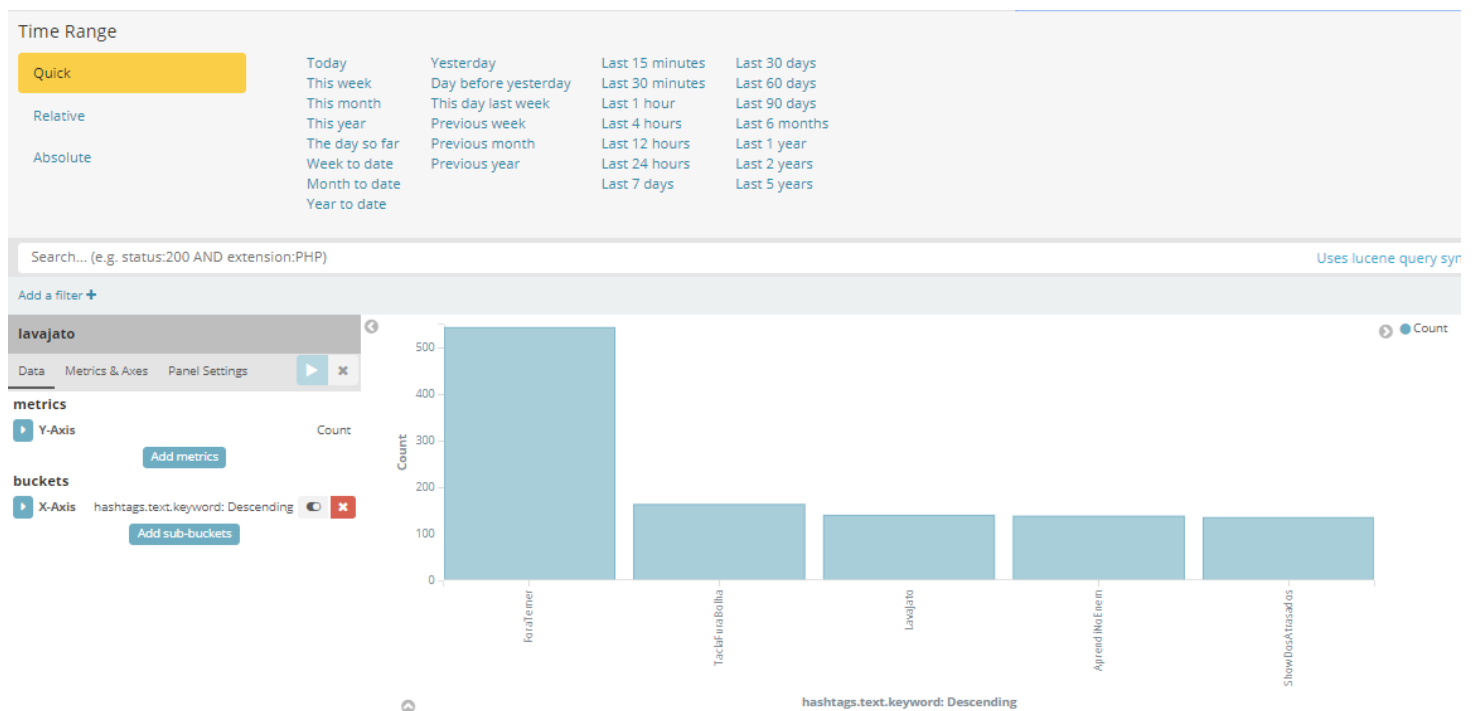


Figura 18 - Escolha do período para visualização da captura de tweets para criação de gráficos

Após os gráficos serem criados e salvos pode-se criar um dashboard com todos e visualizá-los no mesmo ambiente, um ao lado do outro como anteriormente demonstrado na Figura 8.

Para exportar os dados indexados no Elasticsearch pelo Logstash, foi necessário a instalação de um plugin para o Google Chrome chamado Elasticsearch CSV Exporter, desenvolvido pela MineWhat Inc⁷. Até o momento, o trio ELK não possui essa funcionalidade. O arquivo .csv foi utilizado para alimentar o SparkR.

Para o SparkR foi utilizado o arquivo exportado do trio ELK. Primeiramente inicia-se o SparkR pela linha de comando. Na própria linha de comando pode-se utilizar os comandos R, no entanto o presente estudo utiliza o RStudio para melhor visualização.

É necessário iniciar a sessão SparkR no RStudio e baixar o pacote Sparklyr e invocar a biblioteca SparkR. Em seguida um RDD é criado com o arquivo.csv e este filtrado para cada ocorrência de palavra-chave afim de contá-las. Em um segundo momento também foi realizada a criação de gráficos com a biblioteca base do R.

⁷ www.minewhat.com

5 Resultados e Discussão

5.1 Captura de *Tweets*

A captura de *tweets* se deu apenas pelo Elasticsearch e Logstash. O arquivo de configuração do Logstash foi o único arquivo que continha as palavras-chave a serem buscadas. A coleta foi realizada em dois momentos críticos da política nacional. Primeiro quando a ex-presidente Dilma Rousseff sofreu impeachment em agosto de 2016 e o segundo momento durante um dos desdobramentos da Operação Lava Jato em Junho 2017. Entre estas coletas houve coletas para teste no mesmo índice, no entanto somente a última foi utilizada para avaliação.

O SparkR possui uma API de streaming experimental. Pela API não estar estável e estruturada para uso, foi escolhido utilizar os dados extraídos do Elasticsearch mesmo não sendo em tempo real, sendo utilizado no mesmo dia da coleta.

5.2 Utilização do trio ELK

O estudo inicial realizado com o trio ELK demonstrou que este permite, com grande facilidade, a instalação e configuração. Após criação de um arquivo de configuração contendo palavras-chave relacionadas à corrupção e impeachment, foi possível coletar aproximadamente 360224 *tweets* no decorrer de 3 dias seguidos, entre 29 e 30 de Agosto de 2016. A cada dia, foi possível verificar a diferença entre a frequência das palavras chave através do componente Kibana.

As palavras-chave utilizadas foram “foradilma”, “foracunha”, “impeachment”, “naovaitergolpe” (Figura 19). Como pode ser visualizado na

Figura 20, a palavra-chave com mais frequência observada no início do estudo foi “impeachment”, seguida de “anulamaranhao”, “ocupasenado”, “foradilma” e “brazilnocorrupt”. No final da coleta total, a palavra com maior frequência foi “foratemer” seguida de “impeachment”, “impeachmentday”, “pelademocracia” e em quinto “foradilma” (Figura 21).

```
input {
  twitter{
    # add your data
    consumer_key => "
    consumer_secret => "
    oauth_token => "
    oauth_token_secret => "
    keywords => ["foradilma", "foracunha", "impeachment", "naovaitergolpe"]
    #full_tweet => true
  }
}
filter { }
output {
  stdout{
    codec => dots
  }
  elasticsearch{
    hosts => "localhost:9200"
    index => "dilma"
    # index_type => "tweet"
  }
}
```

Figura 19 - Primeira configuração arquivo Logstash

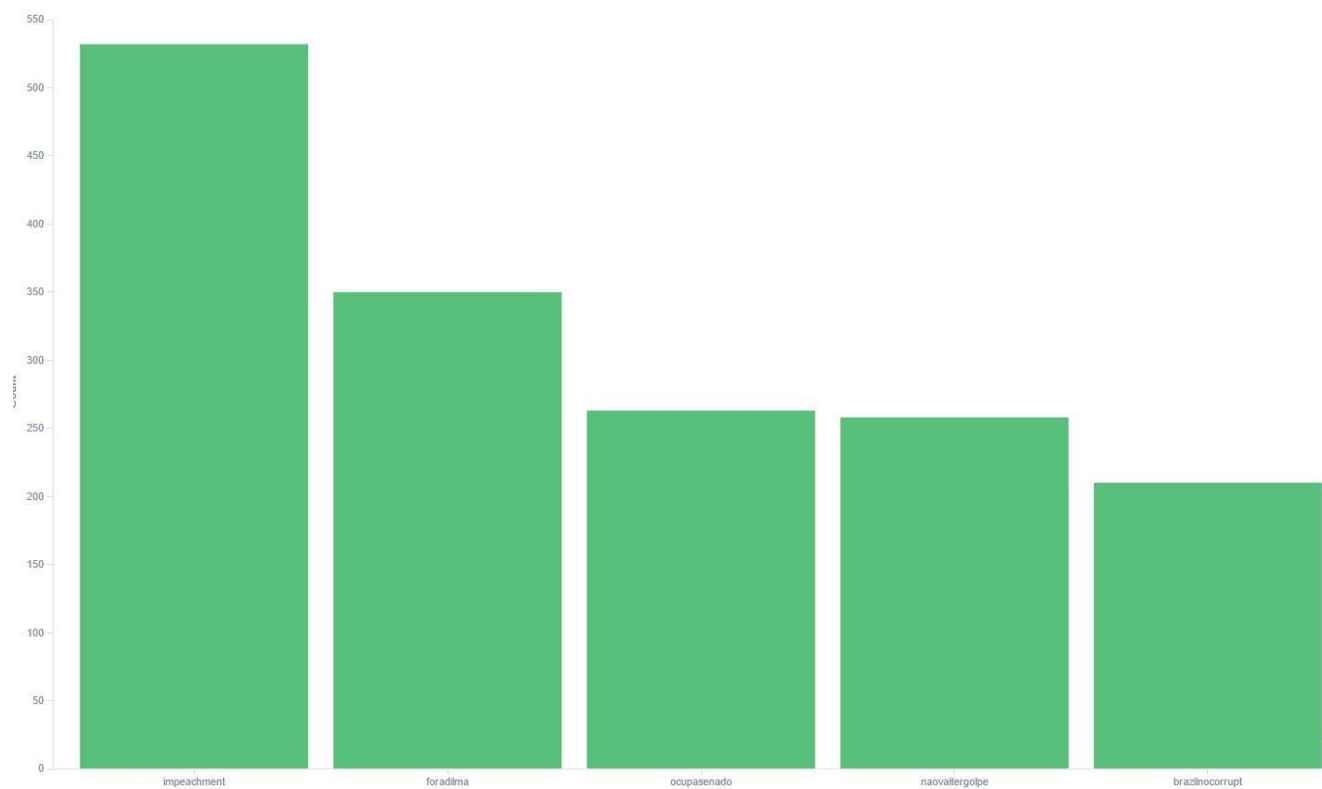


Figura 20 - Frequência palavras-chave do primeiro dia da primeira coleta no trio ELK.

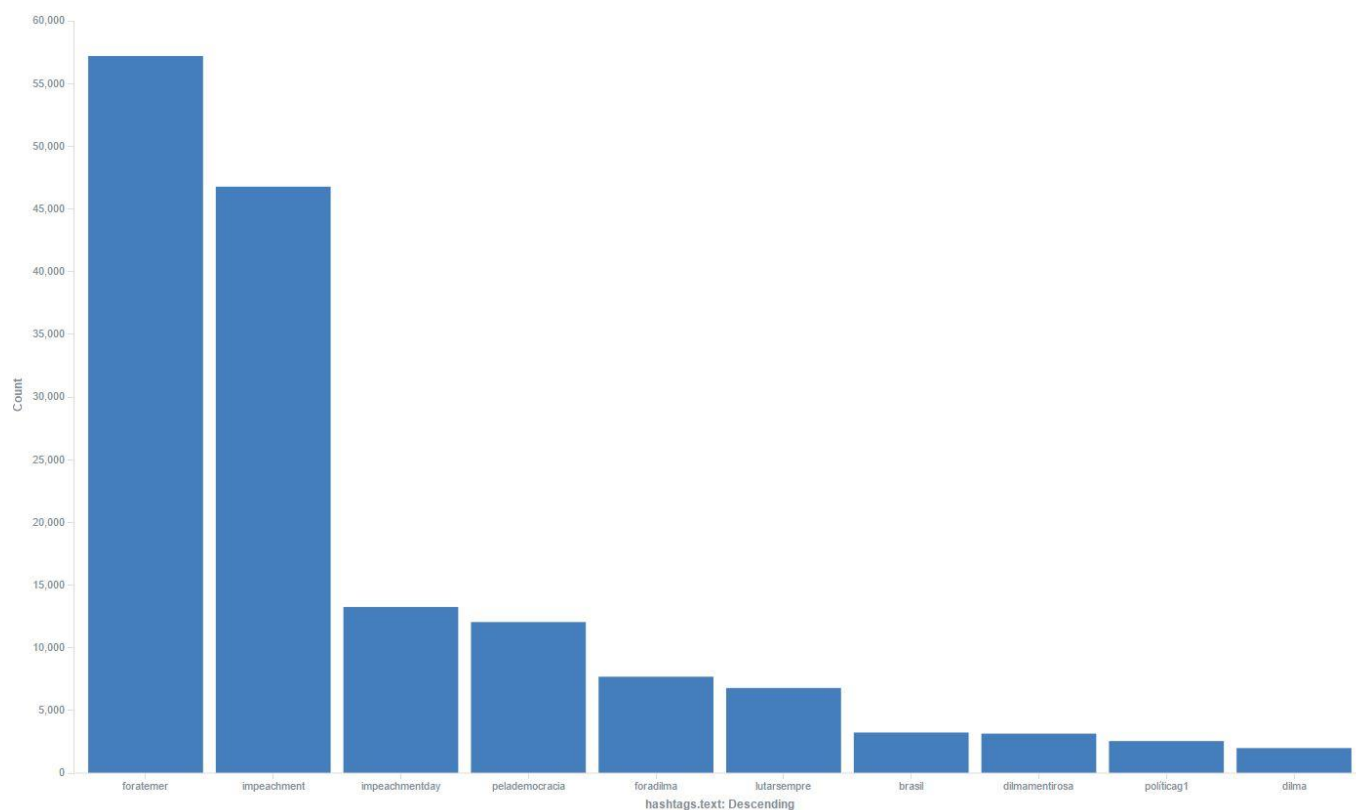


Figura 21 - Dez palavras com maiores frequências ao final da primeira coleta.

hashtags.text: Descending ↕ Q	Count ↕
foratemer	57,192
impeachment	46,773
impeachmentday	13,252
pelademocracia	12,050
foradilma	7,690
lutarsempre	6,792
brasil	3,239
dilmamentirosa	3,151
politicag1	2,557
dilma	2,000

Figura 22 - Quantidade de palavras-chave após primeira coleta.

Na Figura 22 podemos verificar a quantidade de palavras-chave utilizadas nos *tweets* coletados. É possível perceber que não somente as

palavras-chave especificadas no arquivo de configuração são contabilizadas ao usarmos o ELK.

Durante o primeiro momento do presente estudo, foi observado que não somente as palavras-chave configuradas foram coletadas, mas sim todos os *tweets* que continham ao menos uma menção. Isto aumentou muito a quantidade de *tweets* a serem coletados para contar apenas as palavras-chave selecionadas. Houve também quantidade elevada de *tweets* sem palavras-chave, não sendo úteis ao projeto apenas aumentando o tamanho do arquivo coletado. Pode-se notar como o tema alvo da coleta se desenvolveu durante o passar da coleta ao analisar a frequência das palavras-chave. Também é possível notar que as palavras-chave totais no final da coleta não são completamente as mesmas que foram configuradas no arquivo do Logstash. As únicas configuradas que apareceram no resultado foram “foradilma” e “impeachment”. É importante apontar que letras maiúsculas e minúsculas tem impacto no resultado, por exemplo, “Dilma” e “dilma” são tratados diferentemente e reunidos em grupos diferentes.

Num segundo momento, foram coletados aproximadamente 60.000 *tweets* ao longo de três dias, entre 19 e 21 de Julho de 2017. Nessa segunda coleta, as palavras-chave denotavam *hashtags*, ou seja, continham o símbolo jogo da velha (#) como prefixo da mesma forma que aparecem no *Twitter*. As palavras-chave utilizadas foram “#Lulainocente”, “#ForaTemer”, “#impeachment”, “#Impeachment”, “#foratemer”, “#lulainocente”, “#LavaJato”, “#lavajato”, “#LulaNaCadeia”, “#lulanacadeia”, “#Moro”, “#Lula2018”, “#lavajato”.

Apesar da adição do símbolo jogo da velha antes das palavras-chave, foi possível verificar que não somente apenas os *tweets* que as continham foram coletados, mas também diversos *tweets* que não as continham ou não continham palavras-chave (Figura 23). Foi possível perceber também que a ferramenta continuou trazendo e contabilizando palavras-chaves que não estavam contidas no arquivo de configuração como pode-se visto na Figura 24. A inclusão das palavras-chave com variações de letras maiúsculas e minúsculas também foi realizado, no entanto estes continuam sendo reunidos

em grupos diferentes demonstrando que este fato deve ser levado em consideração em qualquer análise feita na ferramenta (Figura 25).

hashtags	message	_index	source	user	retweeted
	RT @Antonio20431788: São farinha do mesmo saco, PT-PMDB uma vergonha. https://t.co/g0sxeSkcDJ	lavajato	http://twitter.com/Antonio20431788/status/927184726439362560	Antonio20431788	false
{ "indices": [29, 39], "text":	November 05, 2017 at 03:45PM #FORATEMER #ELEIÇÕESDIRETAS #CHEGADEGOLPE https://t.co/z8h2Tom8Sd	lavajato	http://twitter.com/oConsciente/status/927184998125404161	oConsciente	false
{ "indices": [86, 108], "text":	Parabéns a todos os trabalhadores e trabalhadoras da nossa indústria cinematográfica! #DiaDoCinemaBrasileiro https://t.co/tdqthGvgzz	lavajato	http://twitter.com/informeRandolfe/status/927188875549831168	informeRandolfe	false
	só posso dizer uma coisa, eu avisei.... https://t.co/MMMk6C	lavajato	http://twitter.com/VickDias4/status/927190305337692160	VickDias4	false

Figura 23 – Palavras-chave faltantes ou não configuradas no arquivo .conf do Logstash

hashtags.text: Descending ↕ Q	Count ↕
lulanacadeia	1,165
foratemer	1,140
democraciacomlula	864
lulainocente	585
lavajato	530
impeachment	363
impeachtrump	360
amjoy	327
trumpsfinaldaysmovietitle	327
lula2018	324

Figura 24 - Quantidade de palavras-chave após segunda coleta



Figura 25 - Aglomeração palavras-chave capturadas

5.3 SparkR

O estudo com o SparkR se deu também em dois momentos da mesma forma que com o trio ELK. No entanto, a biblioteca do SparkR que faria a coleta de dados em tempo real está em modo experimental e, portanto, não foi utilizada no estudo. Para obter os dados, estes foram exportados do Elasticsearch e Logstash utilizando um plugin para o Google Chrome chamado Elasticsearch CSV Exporter desenvolvido pela Minewhat Inc⁸.

Após corretamente inicializada a sessão SparkR no RStudio, o arquivo .csv foi importado para um RDD. A análise inicial contou a ocorrência de cada palavra-chave que tinha sido configurada no arquivo Logstash para o primeiro dia de coleta. Foi possível perceber que mesmo utilizando duas formas diferentes de contagem da palavra-chave, o resultado foi o mesmo. No entanto há diferenças no modo como as palavras-chave podem ser buscadas. Na

⁸ www.minewhat.com

Figura 26, pode-se verificar a diferença de resultados quando a busca é feita apenas pela palavra “impeachment” e pela palavra “\”impeachment\””. A segunda forma é como aparecem as palavras-chave no arquivo que foi exportado (Figura 27), portanto é a forma que procuramos para contagem das mesmas. Nesta ferramenta, assim como no trio ELK, há diferenciação entre letras maiúsculas e minúsculas nas palavras-chave, sendo estas agrupadas separadamente (Figura 28).

```
> dilma <- SparkR:::textFile(sc,'C:/Users/IBM_ADMIN/Desktop/Mine/IC/dilma1.csv')
> count(dilma)
[1] 3494
> impeachment <- SparkR:::filterRDD(dilma, function(line){ grepl("impeachment", line)})
> impeach2 <- count(SparkR:::filterRDD(dilma, function(s) { grepl("impeachment", s) })))
> impeachment2 <- SparkR:::filterRDD(dilma, function(line){ grepl("\\"impeachment\\", line)})
> impeach <- count(SparkR:::filterRDD(dilma, function(s) { grepl("\\"impeachment\\", s) })))
> count(impeachment)
[1] 2153
> paste(impeach2)
[1] "2153"
> count(impeachment2)
[1] 83
> paste(impeach)
[1] "83"
```

Figura 26 - Contagem de palavras-chave do primeiro dia de coletas.

```
[[3397]]
[1] "\"May 7th 2016; 20:32:00.000,@timestamp:May 7th 2016; 20:32:00.000message:RT @radiobandnewsfm: As grades que v\u00f3o dividir o p\u00fablico contr\u00e1rio e favor\u00e1vel ao #impeachment j\u00e1 est\u00e3o no gramado em fre nte ao Congresso Nacional.user:AnnaPSCorreaclient:&lt;a href=\"http://twitter.com/download/android\" rel=\"nofollow\"&gt;Twitter for Android&lt;/a&gt;retweeted:false&source:http://twitter.com/AnnasCorrea/status/729091440882659329hashtags:{ \"text\": \"impeachment\"; \"indices\": [ 82; 94 ] }symbols:user_mentions:{ \"screen_name\": \"radiobandnewsfm\"; \"name\": \"R\u00e1dio B andNews FM\"; \"id\": 26573303; \"id_str\": \"26573303\"; \"indices\": [ 3; 19 ] }@version:1_id:AVSNkDmX4qLkkpmd1_Fm_type:logs_index:dilma_score:\""
```

Figura 27 - Modo de aparição das palavras-chave

```
> anula <- SparkR:::filterRDD(dilma, function(line){ grepl("anulamaramanhao", line)})
> count(anula)
[1] 2
> anula <- SparkR:::filterRDD(dilma, function(line){ grepl("AnulaMaramanhao", line)})
> count(anula)
[1] 124
> anula <- SparkR:::filterRDD(dilma, function(line){ grepl("AnulaMaramanhã", line)})
> count(anula)
[1] 0
> anula <- SparkR:::filterRDD(dilma, function(line){ grepl("anulamaramanhã", line)})
> count(anula)
[1] 0
```

Figura 28 – Diferentes agregações das palavras de acordo com letras maiúsculas e minúsculas

Ao analisarmos o resultado do primeiro dia de coletas no trio ELK e no SparkR conseguimos ver diferenças na contagem. Ao contabilizarmos a palavra “Impeachment”, obtemos valor de 225 ocorrências no trio ELK (Figura 29) enquanto que para o SparkR obtemos 144 ocorrências ao buscarmos pelo modo “\”Impeachment\” e 646 ao buscarmos pelo modo “Impeachment” (Figura 30).

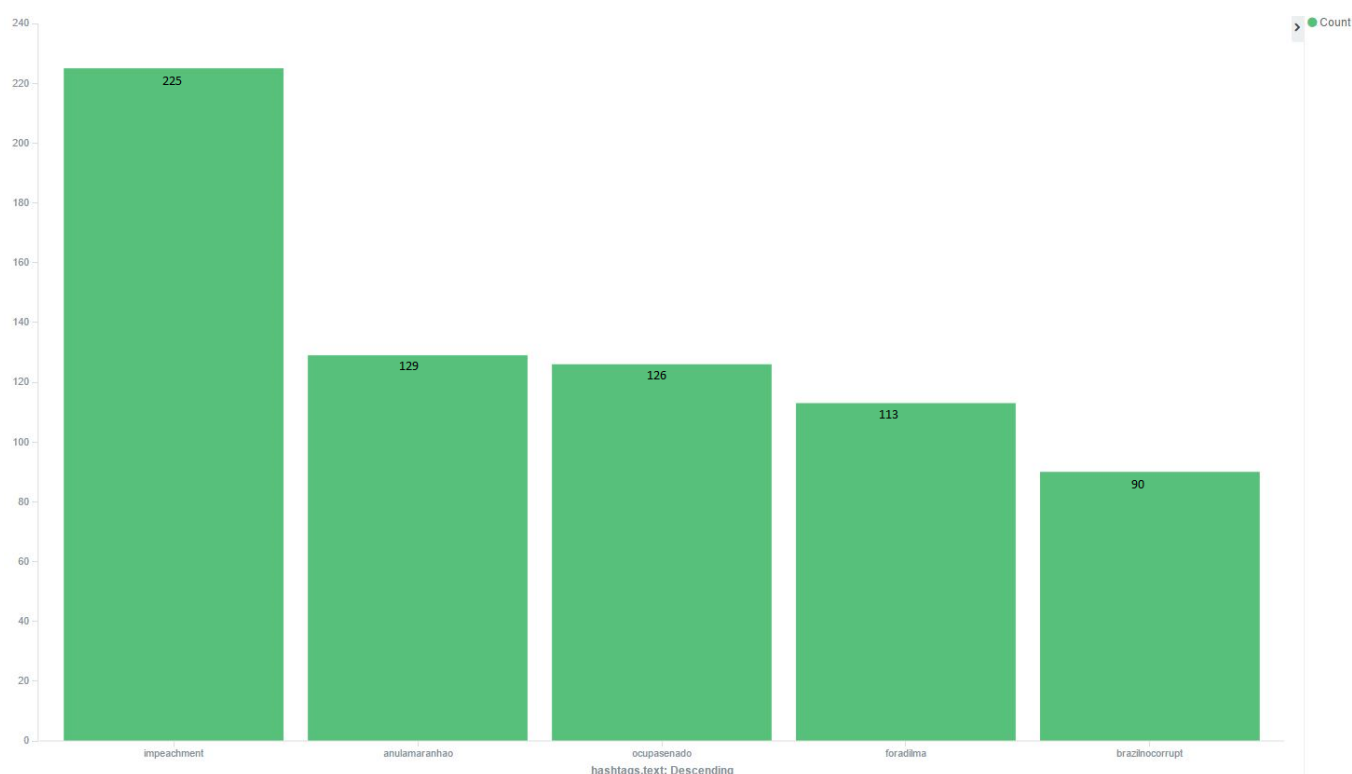


Figura 29 - Número de ocorrências para a palavra "Impeachment" no trio ELK

```
> impeachment <- SparkR:::filterRDD(dilma, function(line){ grepl("Impeachment", line)})
> count(impeachment)
[1] 646
> impeachment <- SparkR:::filterRDD(dilma, function(line){ grepl("\Impeachment\\", line)})
> count(impeachment)
[1] 144
```

Figura 30 - Número de ocorrências para a palavra "Impeachment" e "\Impeachment\" no SparkR

No segundo momento da coleta foi utilizada uma abordagem diferente, na tentativa de igualar os resultados. As palavras-chave foram buscadas como *hashtags*, com o sinal de jogo da velha antecedendo-as. Ao contabilizarmos todas as palavras-chave que foram utilizadas no arquivo de configuração inicial do Logstash obtivemos o resultado da Figura 31. Nesta figura também pode-se verificar que a palavra-chave que apareceu com maior frequência foi “#LulanaCadeia”, diferentemente do resultado quando as palavras foram buscadas sem nenhum tipo de acompanhamento (Figura 32) onde a palavra com maior frequência foi “lavajato”.

```
> count(SparkR::filterRDD(text_file, function(line){ grepl("#LulaInocente", line)}))
[1] 613
> count(SparkR::filterRDD(text_file, function(line){ grepl("#impeachment", line)}))
[1] 212
> count(SparkR::filterRDD(text_file, function(line){ grepl("#Impeachment", line)}))
[1] 81
> count(SparkR::filterRDD(text_file, function(line){ grepl("#foratemer", line)}))
[1] 96
> count(SparkR::filterRDD(text_file, function(line){ grepl("#ForaTemer", line)}))
[1] 947
> count(SparkR::filterRDD(text_file, function(line){ grepl("#lulainocente", line)}))
[1] 32
> count(SparkR::filterRDD(text_file, function(line){ grepl("#LavaJato", line)}))
[1] 602
> count(SparkR::filterRDD(text_file, function(line){ grepl("#lavajato", line)}))
[1] 31
> count(SparkR::filterRDD(text_file, function(line){ grepl("#LulanaCadeia", line)}))
[1] 1067
> count(SparkR::filterRDD(text_file, function(line){ grepl("#lulanacadeia", line)}))
[1] 50
> count(SparkR::filterRDD(text_file, function(line){ grepl("#moro", line)}))
[1] 51
> count(SparkR::filterRDD(text_file, function(line){ grepl("#Moro", line)}))
[1] 308
> count(SparkR::filterRDD(text_file, function(line){ grepl("#Lula2018", line)}))
[1] 262
> count(SparkR::filterRDD(text_file, function(line){ grepl("#lula2018", line)}))
[1] 23
```

Figura 31 - Frequência de palavras-chave precedidas do símbolo jogo da velha no SparkR

```

> text_file <- SparkR:::textFile(sc, 'C:/Users/IBM_ADMIN/Desktop/Mine/Unirio/IC/LavaJato.csv')
> count(text_file)
[1] 56771
> fora <- SparkR:::filterRDD(text_file, function(line){ grepl("ForaTemer", line)})
> count(for_a)
[1] 1079
> count(SparkR:::filterRDD(text_file, function(line){ grepl("LulaInocente", line)}))
[1] 613
> count(SparkR:::filterRDD(text_file, function(line){ grepl("impeachment", line)}))
[1] 7745
> count(SparkR:::filterRDD(text_file, function(line){ grepl("Impeachment", line)}))
[1] 1690
> count(SparkR:::filterRDD(text_file, function(line){ grepl("foratemer", line)}))
[1] 100
> count(SparkR:::filterRDD(text_file, function(line){ grepl("ForaTemer", line)}))
[1] 1079
> count(SparkR:::filterRDD(text_file, function(line){ grepl("lulainocente", line)}))
[1] 34
> count(SparkR:::filterRDD(text_file, function(line){ grepl("LavaJato", line)}))
[1] 809
> count(SparkR:::filterRDD(text_file, function(line){ grepl("LavaJato", line)}))
[1] 56771
> count(SparkR:::filterRDD(text_file, function(line){ grepl("LulanaCadeia", line)}))
[1] 1067
> count(SparkR:::filterRDD(text_file, function(line){ grepl("lulanacadeia", line)}))
[1] 50
> count(SparkR:::filterRDD(text_file, function(line){ grepl("moro", line)}))
[1] 8753
> count(SparkR:::filterRDD(text_file, function(line){ grepl("Moro", line)}))
[1] 29765
> count(SparkR:::filterRDD(text_file, function(line){ grepl("Lula2018", line)}))
[1] 291
> count(SparkR:::filterRDD(text_file, function(line){ grepl("lula2018", line)}))
[1] 27

```

Figura 32 - Frequência de palavras-chave no SparkR

Quando comparamos os resultados, o que mais se aproxima do resultado obtido pelo trio ELK é o com uso da *hashtag*, uma vez que a palavra-chave com maior ocorrência no trio ELK foi “LulanaCadeia” com 1165 ocorrências (Figura 24). Com esta abordagem os resultados das ferramentas se aproximaram, diferentemente da primeira abordagem.

Um gráfico foi plotado com uma função simples do R chamada *barplot*, utilizando o arquivo da segunda coleta (Figura 33). Este gráfico é inferior ao gráfico criado com o trio ELK (Figura 34) e com maior número de ações para criá-lo. Enquanto o trio ELK possui uma interface intuitiva e proporciona criação de gráficos e imagens com facilidade, o SparkR necessita de noções de programação e mais comandos para produção de um gráfico que utiliza os mesmos princípios do criado pelo trio ELK. No entanto, o SparkR proporciona a possibilidade de escolher quais palavras-chave estarão presentes no gráfico no momento que este é criado enquanto o trio ELK permite apenas escolher quantas palavras-chave irão compor o gráfico e se serão em ordem crescente ou decrescente. Após o gráfico ser gerado, pode-se escolher por meio de filtros

se uma palavra-chave deve ser retirada ou não do gráfico, da mesma forma que pode-se escolher apenas aquelas que devem estar presentes.

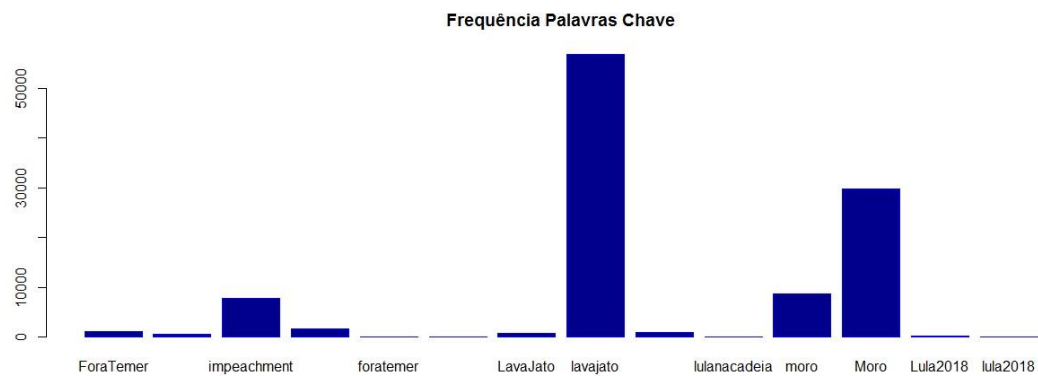


Figura 33 - Frequência de palavras-chave na segunda coleta no SparkR

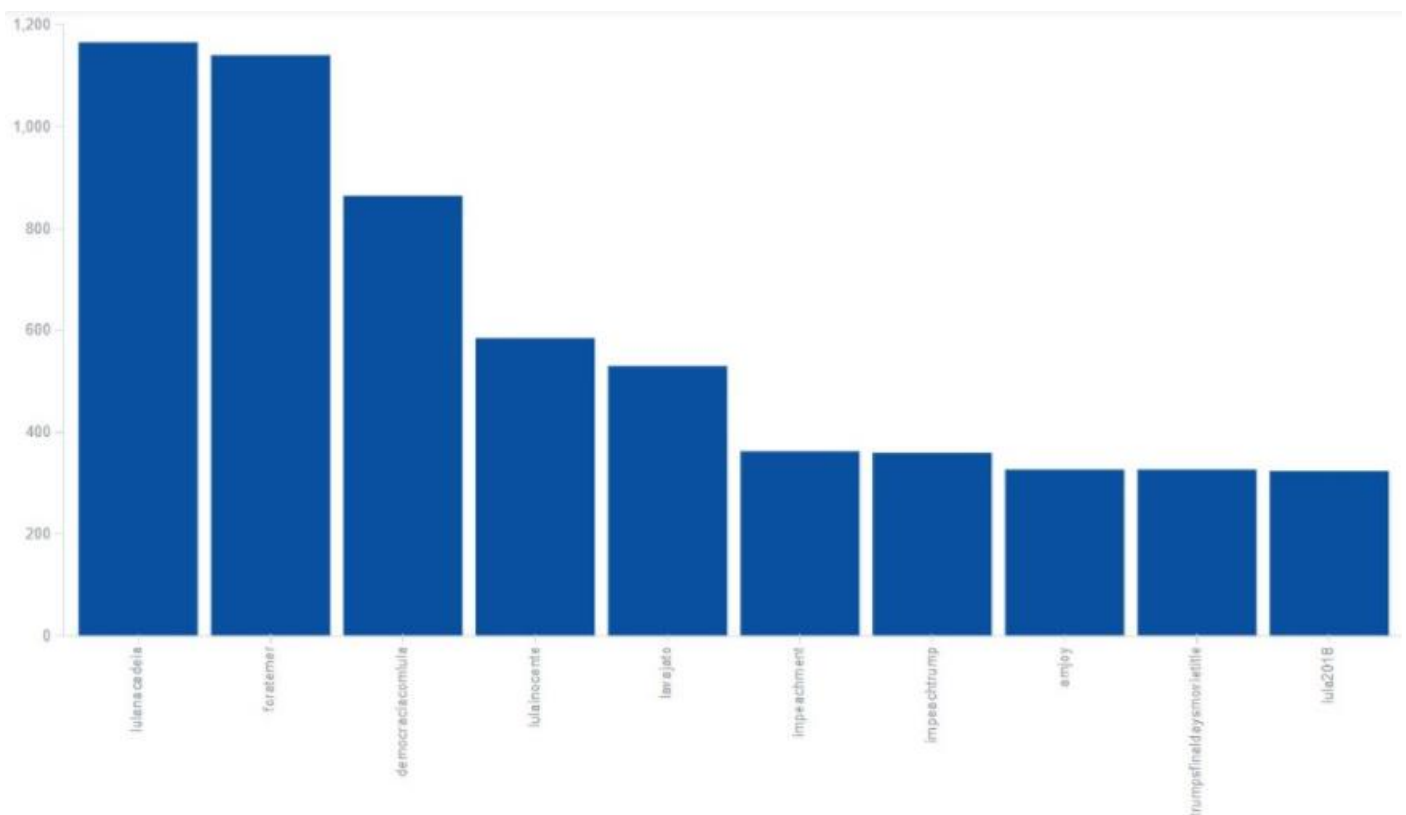


Figura 34 - Frequência de palavras-chave na segunda coleta no trio ELK

5.4 Discussões sobre o estudo

A partir do estudo realizado foi possível conhecer diversas ferramentas e aprofundar conhecimentos em algumas delas, assim como nas áreas de mineração de dados e computação cognitiva.

A partir das análises realizadas com as ferramentas escolhidas foi possível perceber a importância dos arquivos de configuração das ferramentas assim como as palavras-chaves escolhidas e como estas são escritas para a busca. Durante as análises foi possível verificar que letras maiúsculas e minúsculas eram importantes ao compor as palavras-chave para a busca no *Twitter* uma vez que as palavras-chaves eram coletadas e agregadas exatamente como estavam escritas no arquivo de configuração. A busca também retornou outras palavras-chave que não estavam presentes no arquivo de configuração uma vez que a maioria dos usuários utilizam mais de uma palavra-chave por *tweet*. O trio ELK mostra as palavras-chave com maior frequência, independentemente de ter sido configurada no arquivo de configuração do Logstash, levando à *insights* para a pesquisa podendo ser tratado como computação cognitiva. O SparkR mostra apenas a contagem das palavras-chave que são especificadas no comando de contagem, sendo difícil analisar a frequência de palavras-chaves não configuradas, o que pode ser entendido como uma desvantagem da ferramenta, assim como demonstrativo da falta de cognição na mesma.

Ao utilizar as ferramentas escolhidas foi possível visualizar como a mineração em tempo real funciona e se os dados coletados refletem o momento da coleta, sendo possível ajustar a busca para melhor retratar o cenário sendo analisado. Cada ferramenta possui um método diferente de busca e visualização de dados, sendo o SparkR a mais complexa das duas pois há necessidade de saber a linguagem de programação R para analisar e visualizar os resultados. O trio ELK não necessita de prévio conhecimento de programação para visualizar as buscas, sendo fácil e intuitiva tanto a criação dos gráficos como adicionar e remover filtros da visualização, assim como limitar o período de tempo a ser analisado ou demonstrado no gráfico. A Tabela 1 demonstra as características de cada ferramenta.

Característica	Trio ELK	SparkR
Presença de palavras-chave não configuradas	Sim	Sim
Necessita programação básica	Não	Sim
<i>Insights</i> de palavras-chave não buscadas	Sim	Não
Criação de gráficos e filtros com facilidade	Sim	Não
Busca em tempo real no Twitter	Sim	Não
Possibilidade de exportação dos dados coletados	Não	Sim

Tabela 2 - Quadro comparativo das ferramentas Trio ELK e SparkR

5.5 Limitações do Estudo

O estudo teve como limitação a não obtenção de *tweets* em tempo real pela ferramenta SparkR uma vez que a biblioteca que faria esta coleta está em modo experimental. Outra limitação foi a impossibilidade de exportação do resultado da coleta pelo trio ELK sendo necessário um *plugin* externo à ferramenta para exportar os dados obtidos. Este *plugin* era capaz de exportar apenas 500 *tweets* por vez tornando a exportação manual, longa e passível de erro humano. A primeira coleta do estudo retornou 360224 *tweets*, que devido à limitação de exportação, não foi possível ser analisada por completo no SparkR. Apesar desta limitação, foi possível comparar o primeiro dia da coleta do primeiro momento entre o trio ELK e SparkR e toda a coleta do segundo momento.

6 Conclusão

Este trabalho apresentou estudo de ferramentas de análise de dados em tempo real, um dos desafios da computação cognitiva. Foram coletados *tweets* em dois momentos distintos em tempo real e analisados nas ferramentas ELK e SparkR.

6.1 Considerações Gerais

Inicialmente ambas ferramentas seriam utilizadas com captura dos *tweets* em tempo real, no entanto a biblioteca do SparkR para este tipo de coleta ainda é experimental, portanto foi deixada de fora do estudo. A coleta se deu apenas pelo Logstash utilizando o arquivo de configuração que continha as palavras-chaves a serem buscadas. Foi possível reparar que ao trazer os *tweets* que continham as palavras-chave selecionadas, outras palavras-chave também apareciam na busca uma vez que o usuário pode escrever mais de uma palavra-chave por *tweet*. A língua de coleta não foi restrita, uma vez que a cena política brasileira atingiu cobertura internacional. As ferramentas escolhidas não são as únicas que podem ser utilizadas como parte da computação cognitiva, sendo estas escolhidas após estudo via artigos e facilidade de instalação.

As coletas foram feitas em dois momentos, ambos com duração de 3 dias ininterruptos e em momentos críticos da cena política no Brasil. Foi possível perceber no primeiro momento da coleta que, ao utilizar apenas palavras no arquivo de configuração, as mais diversas palavras-chaves foram coletadas. Foi possível verificar também que a primeira coleta obteve mais *tweets* que a segunda tendo aproximadamente 6 vezes mais ocorrências.

6.2 Trio ELK

Ambas ferramentas realizaram bem o seu propósito, no entanto o trio ELK se mostrou uma ferramenta mais completa para computação cognitiva. Esta é mais intuitiva e fácil de usar, além de mostrar os resultados em tempo real em um dashboard bem estruturado e de acordo com o propósito da pesquisa. Há possibilidade também de filtrar o que aparecerá ou não nos gráficos de forma fácil sem a necessidade de criação de outro gráfico. E caso não deseje mais o filtro, pode-se retirá-lo apenas com um clique. O arquivo de configuração do Logstash pode ser facilmente ampliado e mais específico para cada tipo de dados e tipos de alimentação necessários. Não há necessidade de saber programação a fundo, mas caso a intenção seja utilizar o trio de forma mais complexa, o mínimo de programação se faz necessário. O Logstash permite alimentação de diversas fontes, unificá-las, indexar em diversos locais e ferramentas e exportá-las para locais diversos. Uma limitação da ferramenta é a não possibilidade de exportar os resultados dos dados coletados de maneira nativa, sendo preciso um plugin para o navegador Google Chrome que apenas exporta 500 ocorrências por vez. Esta limitação da ferramenta também limitou a análise do trabalho uma vez que a primeira etapa da coleta houve 360224 ocorrências, o que tornou extremamente longa a exportação e passível de erros.

6.3 SparkR

A ferramenta Spark, mais precisamente seu pacote SparkR, foi desenvolvido para utilizar o Spark no R. O SparkR fornece uma implementação de RDD que suporta operações como seleção, filtragem e agregação em grandes conjuntos de dados. Diferentemente do trio ELK, é necessário o conhecimento em programação em uma das linguagens suportadas (Scala, Python, Java, R). Foi verificado também que, apesar de haver uma documentação bem estruturada na própria página da ferramenta, esta não abrange todas as funcionalidades. Infelizmente não foi possível utilizar a função streaming inclusa no Spark nem o novo mecanismo de processamento de fluxo para SparkR, o que fez o estudo desviar um pouco do objetivo inicial, sendo uma limitação do projeto. A criação de gráficos e análise de dados não é tão

intuitiva quanto o trio ELK, sendo mais demorado para criação e por vezes havendo necessidade de instalação de bibliotecas extras. Caso o gráfico não saia como o desejado, como por exemplo, palavras ou termos que deveriam ter sido filtrados, há necessidade de recriá-lo do zero. Foi verificado também que dependendo de como a palavra-chave é configurada para ser localizada no RDD, o resultado varia, diferentemente do trio ELK.

O melhor entendimento do Spark em si e do SparkR e como uni-los com outras ferramentas será essencial para utilizá-los para captura em tempo real para uma nova fase da pesquisa. Pode-se também utilizar o Spark sem a interface R como um ramo do estudo.

No âmbito da computação cognitiva, ambas ferramentas são úteis pois lidam com enormes quantidades de dados de modo rápido e seguro além de possuírem interfaces com outras ferramentas utilizadas na computação cognitiva. Unir as ferramentas atuais com outras ferramentas para computação cognitiva é outra proposta para pesquisa.

6.4 Trabalhos Futuros

Como trabalhos futuros pode-se citar a inclusão do Spark sem a interface R juntamente com sua biblioteca Streaming para busca dos *tweets* em tempo real assim como ocorre no trio ELK. Modificar a configuração do Logstash também pode levar a uma análise mais eficiente uma vez que os *tweets* seriam mais limitados ao objetivo da busca. Uma vez que ambas ferramentas proporcionam interface com outras ferramentas utilizadas na computação cognitiva, uni-las e abranger o modo como os dados são analisados também poderia tornar a análise mais eficiente e ligada ao objetivo inicial.

Referências Bibliográficas

- [1] J. Manyika e e. al, “Big data: The next frontier for innovation, competition, and productivity,” McKinsey, Maio 2011. [Online]. Available: <https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/big-data-the-next-frontier-for-innovation>. [Acesso em 5 Novembro 2017].
- [2] EMC Education Services, “Introduction to Big Data Analytics,” em *Data Science & Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data*, Indianapolis, IN, USA, John Wiley & Sons, Inc, 2015, pp. 18-42.
- [3] C. Snijders, U. Matzat e U.-D. Reips, ““Big Data”: Big Gaps of Knowledge,” *International Journal of Internet Science*, vol. 7, nº 1, p. 1–5, 2012.
- [4] A. A. Safaei, “Real-time processing of streaming big data,” *Real-Time Syst*, vol. 53, p. 1–44, 2016.
- [5] J. Kelley III, “Computing, cognition, and the future of knowing: how humans and machines are forging a new age of understanding.,” Outubro 2015. [Online]. Available: http://www.research.ibm.com/software/IBMResearch/multimedia/Computing_Cognition_WhitePaper.pdf. [Acesso em 21 Outubro 2017].
- [6] Deloitte, “Artificial intelligence, real results.,” 2015. [Online]. Available: <https://www2.deloitte.com/us/en/pages/about-deloitte/articles/gx-artificial-intelligence-cognitive-computing.html>. [Acesso em 21 Outubro 2017].
- [7] S. TK e R. Viswanathan, “Cognitive Computing: The Next Stage in Human/Machine Coevolution,” Cognizant, Abril 2017. [Online]. Available: <https://www.cognizant.com/whitepapers/cognitive-computing-the-next-stage-in-human-machine-coevolution-codex2582.pdf>. [Acesso em 20 Novembro 2017].
- [8] W. Wingerath, F. Gessert, S. Friedrich e N. Ritter, “Real-time stream processing for Big Data.,” *it - Information Technology*, vol. 58, nº 4, pp. 186-194, 2016.
- [9] M. Blum e G. Schuh, “Towards a Data-oriented Optimization of Manufacturing Processes,” *Proceedings of the 19th International Conference on Enterprise Information Systems*, pp. 257-264, 2017.
- [10] H. Packard, “BRAIN: Neuromorphic computing and the future of artificial cognition.,” [Online]. Available: <https://www.labs.hpe.com/next-next/brain>. [Acesso

- em 21 Outubro 2017].
- [1] KPMG, “Embracing the cognitive era,” 2016. [Online]. Available:
 - 1] <https://assets.kpmg.com/content/dam/kpmg/pdf/2016/03/embracing-the-cognitive-era.pdf>. [Acesso em 21 Outubro 2017].
 - [1] Y. Chen, E. Argentinis e G. Weber, “IBM Watson: How Cognitive Computing Can
 - 2] Be Applied to Big Data Challenges in Life Sciences Research,” *Clinical Therapeutics*, vol. 38, n° 4, pp. 688-701, 2016.
 - [1] R. Hull e H. Motahari Nezhad , “Rethinking BPM in a Cognitive World:
 - 3] Transforming How We Learn and Perform Business Processes,” *Business Process Management Conference, Rio de Janeiro*, vol. 9850, pp. 3-19, 2016.
 - [1] R. High, “The Era of Cognitive Systems: An Inside Look at IBM Watson and How
 - 4] It Works.,” IBM, Dezembro 2012. [Online]. Available: www.redbooks.ibm.com/abstracts/redp4955.html?open. [Acesso em 22 Outubro 2017].
 - [1] F. Santoro e F. Baião, “Knowledge-intensive Process: A Research Framework,” em
 - 5] *1st Workshop on Cognitive Business Process Management in conjunction with International Conference on Business Process Management*, Barcelona, 2017.
 - [1] S. Wasserman e K. Faust, *Social Network Analysis: Methods and Applications*,
 - 6] Cambridge University Press., 1994.
 - [1] Y. Gu, Z. (. Qian e F. Chen, “From Twitter to detector: Real-time traffic incident
 - 7] detection using social media data,” *Transportation Research Part C: Emerging Technologies*, vol. 67, pp. 321-342, 2016.
 - [1] “OAuth Domentação,” [Online]. Available: <https://oauth.net/>. [Acesso em 02
 - 8] November 2017].
 - [1] “What is open source,” [Online]. Available: [https://opensource.com/resources/what-](https://opensource.com/resources/what-open-source)
 - 9] [open-source](https://opensource.com/resources/what-open-source). [Acesso em 30 October 2017].
 - [2] J. Gay, *Free Software, Free Society: Selected Essays of Richard M. Stallman*,
 - 0] Boston: Free Software Foundation, 2002.
 - [2] Elastic, “Documentação Elastic,” [Online]. Available:
 - 1] <https://www.elastic.co/guide/index.html>. [Acesso em 22 Outubro 2017].
 - [2] A. Tost e J. d. Jesús, “Escalabilidade e elasticidade para padrões de aplicativo

- 2] virtual no IBM PureApplication System,” IBM, 01 Novembro 2013. [Online]. Available:
https://www.ibm.com/developerworks/br/websphere/techjournal/1309_tost/index.html. [Acesso em 20 Novembro 2017].
- [2 Hadoop, “Apache Hadoop,” Apache, [Online]. Available: <http://hadoop.apache.org>.
3] [Acesso em 20 Novembro 2017].
- [2 Apache, “Documentação Spark,” [Online]. Available:
4] <http://spark.apache.org/docs/latest/index.html>. [Acesso em 5 Novembro 2017].
- [2 Apache, “Documentação SparkR,” [Online]. Available:
5] <http://spark.apache.org/docs/latest/sparkr.html#overview>. [Acesso em 05 Novembro 2017].
- [2 R, “Documentação R,” [Online]. Available: <https://www.r-project.org/about.html>.
6] [Acesso em 05 Novembro 2017].
- [2 “FAQ R,” [Online]. Available: https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-R_003f. [Acesso em 05 Novembro 2017].
- [2 Apache, “Apache Spark,” [Online]. Available:
8] <https://spark.apache.org/docs/2.2.0/rdd-programming-guide.html#resilient-distributed-datasets-rdds>. [Acesso em 20 Novembro 2017].
- [2 P. Pääkkönen e D. Pakkala, “Reference Architecture and Classification of
9] Technologies, Products and Services for Big Data Systems.,” *Big Data Research*, vol. 2, nº 4, p. 166–186, 2015.