UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA
DEPARTAMENTO DE ESTATÍSTICA

Text Mining utilizando o *Software* R: um estudo de caso de uma biblioteca americana

Autor: Jorge Luiz Staudt Junior

Orientadora: Professora Dra. Lisiane Priscila Roldão Selau

Porto Alegre, Julho de 2016.

Universidade Federal do Rio Grande do Sul
Instituto de Matemática
Departamento de Estatística

Text Mining utilizando o *Software* R: um estudo de caso de uma biblioteca americana

Autor: Jorge Luiz Staudt Junior

Trabalho de Conclusão de Curso
apresentado para obtenção
do grau de Bacharel em Estatística.

Banca Examinadora:
Professora Dra. Lisiane Priscila Roldão Selau (orientadora)
Professor Dr. Guilherme Pumi

Porto Alegre, Julho de 2016.

Dedico este trabalho à minha família.

“You, me, or nobody is gonna hit as hard as life. But it ain't about how hard you hit. It's about how hard you can get hit and keep moving forward. How much you can take and keep moving forward. That's how winning is done!”
(Rocky Balboa)

Agradecimentos

Primeiramente agradeço a Deus.

À minha família, Jorge, Katia, Priscilla e Aline, pelo amor, companheirismo, ajuda, incentivo e compreensão. Obrigado por me ensinarem a nunca desistir dos meus sonhos, a buscar novas experiências, ser honesto, íntegro e uma pessoa de bom coração. Agradeço aos meus pais, Jorge e Katia por todo suporte para entrar e completar minha graduação e o grande amor por mim, à minha irmã mais velha Priscilla pelos diversos conselhos e a caçula Aline pelas brincadeiras para me deixar sempre animado. Amo vocês.

Aos meus cachorros, Billy, Pitoco e Thor pelas demonstrações de carinho, por me ensinarem o valor da amizade e por aguentarem minhas brincadeiras pacientemente. Que Deus cuide de Billy e Pitoco, e que conceda uma vida longa para o Thor.

Às minhas avós, Alcina e Dorothy, pessoas que sempre me incentivaram nessa trajetória da faculdade, mas que por uma peça do destino não poderão ver de perto essa conclusão.Guardarei sempre no coração as boas lembranças juntos, os incentivos para os estudos e os conselhos para a vida.Saudades.

A todos meus amigos que compartilham momentos bons e ruins comigo, nos quais citarei alguns nomes: Claiton, Diego, Leandro, Leonardo e Pablo. Aos meus colegas Allan, Douglas, Felipe, Lucas e Thiago pelo companheirismo, ajuda nas matérias, rodas de chimarrão e risadas.

A todos os professores de Estatística que me ensinaram muito e me aturaram em todos esses anos. Principalmente Flávio e Patrícia, por seus diversos conselhos e porque foi a partir deles que quis ingressar na faculdade de Estatística. Também agradeço ao Professor Álvaro pela oportunidade de estudar no exterior durante a graduação.

À minha orientadora, Lisiane, pelos conselhos diversos, pelas diversas cadeiras lecionadas na faculdade que me incentivaram muito a continuar fazendo Estatística, pela compreensão dos meus atrasos nas aulas e por me orientar mesmo sem tempo, me ajudando sempre que necessário.

Resumo

A quantidade de dados textuais existente na rede de computadores é enorme, pois muitas pessoas e empresas usam a *internet* diariamente para expressarem suas opiniões sobre diversos assuntos. Esses dados textuais podem conter informações valiosas, que muitas vezes, podem ser obtidas com rapidez e baixo custo financeiro, como a informação obtida nas redes sociais. Nas redes sociais são estimadas milhares de postagens de escrita e fotos por segundo. Sendo assim, o domínio de técnicas para extrair informações de bases textuais sem necessidade de leitura prévia é de grande relevância. Tendo em vista a busca de informações pertinentes e relevantes, um programa de leitura da biblioteca da cidade de Chicago, decidiu usar a técnica de *Text Mining* para extrair essas informações na rede social *Twitter*, em busca de ideias para aperfeiçoamento e continuidade do programa. Hoje em dia, há diversos *softwares* pagos e gratuitos que contém a técnica do *Text Mining*. Dessa maneira, o objetivo desse trabalho é estudar o processo de *Text Mining* desde a obtenção até a análise dos dados e seu uso no *Software* R, além de mostrar sua aplicação para ajudar o programa de leitura de Chicago a obter as informações para tomada de decisões. Com o *Software* R, foram coletados *tweets* sobre o programa de leitura da biblioteca de Chicago. Inicialmente, esses dados foram devidamente preparados para análise, depois foram construídos gráficos de frequências e nuvem de palavras. Para dividir os *tweets* por assunto foram utilizadas três diferentes técnicas de *Clustering* e modelagem por tópicos. Com a limpeza e análise dos *tweets* foi possível obter uma ideia dos diversos assuntos que as pessoas estavam falando no *Twitter* sobre o programa de leitura.

Palavras-chave: *Text Mining, Software* R, *Clustering, Twitter*.

Sumário

1. INTRODUÇÃO

Text Mining, também conhecido como "*Document Mining*" é o processo de obtenção de informações importantes de bases textuais não estruturadas. Pode também ser visto como uma extensão do *Data Mining*, que é a extração de conhecimento de bases de dados estruturadas. É considerado um campo multidisciplinar, envolvendo recuperação de informação, análise de texto, extração de informação, *clustering*, classificação, visualização, banco de dados tecnológicos, *machine learning* e *data mining* (TAN, 1999).

Atualmente muitas pessoas e empresas usam a *internet* para expressarem suas opiniões sejam sobre pessoas, produtos, eventos, entre outros. Com isso a quantidade de dados textuais que existe na rede de computadores é enorme e esses dados podem ser usados a favor de melhorias. Sendo assim, a informação textual é muitas vezes um caminho pelo qual se pode ter informação valiosa e de graça, visto que a informação encontrada nas redes sociais é de livre acesso. O benefício do *Text Mining* se dá pela grande quantidade de informação valiosa contida nos textos e que não está disponível nos dados estruturados clássicos (FEINERER, 2008).

Segundo o site *olhar digital da* UOL (usando dados do infográfico desenvolvido pela Qmee) estima-se que no *Facebook* são feitas 41 mil postagens por segundo, no *Twitter* 278 mil *tweets* por segundo, 204 mil emails são enviados por minuto e 20 mil postagens são feitas no *Tumblr* por segundo. Contabilizando esses dados por semana ou mês pode-se chegar a números que ultrapassam bilhões de postagens e que representam dados relevantes para obter informação de diversos locais, produtos, pessoas ou empresas. Segundo Tan (1999), mais da metade da informação de uma companhia está armazenada em dados não estruturados. Isso não quer dizer que se deve dar atenção somente a dados não estruturados, e sim que as empresas devem olhar para eles como uma fonte de grande informação.

Um exemplo de aplicação do *Text Mining* é seu uso em um programa de leitura da cidade de Chicago "*One Book One Chicago program* – OBOC". O programa tem como propósito envolver os residentes da cidade de Chicago e promover um senso de comunidade por meio da leitura. Com o objetivo de identificar possibilidades de melhorias, os gestores do programa

decidiram, a partir de dados coletados do *Twitter*, extrair informações relevantes para que possam pensar em ideias de aperfeiçoamento e continuidade do programa.

Atualmente existe uma grande quantidade de *softwares* para *Text Mining*, sendo muitos deles de uso gratuito. Os de uso gratuito não deixam a desejar em relação aos pagos no que diz respeito à qualidade, tendo muitos recursos para atender às demandas de análise dos usuários. O site "*predictive analytics today*" aponta como os vinte e três melhores *softwares* gratuitos disponíveis hoje na internet, dentre eles: *QDA Miner Lite, KH Coder, TAMS Analyzer, Carrot2, CAT e R*. Dentre os *softwares* pagos com análise de *Text Mining*, os que mais se destacam são *SAS Text Analytics* e IBM *Text Analytics*. Mais especificamente, o *Software* R possui boas ferramentas de análise como o pacote '*tm'* que possui uma introdução do *Text Mining*, '*wordcloud'* para fazer nuvem de palavras, '*RColorBrewer'* para cores de mapas temáticos e '*fpc'* para auxílio de métodos, validação e estimação de *clusters.*

Nesse sentido, o presente trabalho possui como objetivo principal estudar a técnica do *Text Mining* e seu uso através do *Software R*, e como objetivo secundário utilizar a técnica de *Text Mining* para analisar os *tweets* coletados pelo programa *One Book One Chicago*, de forma a obter informação relevante para melhoria e continuidade do programa.

A divisão do trabalho está feita em cinco seções, sendo esta a primeira seção. A segunda seção vai apresentar o que é *Text Mining*, como esse procedimento é feito passo a passo, as novidades de uso da técnica e exemplos de aplicação. A terceira seção trata da metodologia, primeiramente apresentando o *Software R*, que será utilizado para obter os resultados da quarta seção, depois descrevendo a aplicação do *Text Mining* para o programa *One Book One Chicago* com os dados e etapas que serão utilizados. A quarta seção conta com a aplicação do problema dado pelo programa *One Book One Chicago* com dados do *Twitter*, mostrando os passos da análise e os resultados obtidos. Na última seção constarão as conclusões deste trabalho e suas limitações.

2. *TEXT MINING*

Durante os últimos anos o *Text Mining* vem sendo amplamente utilizado, tendo como base a estatística e o aprendizado de máquina, sendo o texto a informação de entrada (FEINERER, 2008).

Segundo Dixon (1997) e Morais e Ambrósio (2007), o processo de *Text Mining* se divide em quatro etapas:

1. Recuperação da Informação/ Identificação do Problema: o primeiro passo é localizar os documentos com informações relevantes para o assunto que está sendo estudado, filtrar esses documentos, permanecendo somente aqueles com as informações necessárias.
2. Extração da Informação/ Pré- Processamento: a próxima etapa é extrair informações dos documentos selecionados. Essa extração é tipicamente um processo de estruturar os dados.
3. Mineração da Informação/ Mineração dos dados: uma vez que os textos selecionados foram transformados para dados estruturados, entra-se em um estágio em que os dados ficam compatíveis para o uso de técnicas de *Data Mining*. Nessa etapa tenta-se descobrir padrões nos dados.
4. Interpretação/ Pós- Processamento: o último passo é interpretar os padrões descobertos na fase de mineração. Essa interpretação é feita usualmente com linguagem natural.

A etapa de Extração da informação é uma dos problemas chave do *Text Mining*, pois serve como ponto de partida para os algoritmos de análise (AGGARWAL; ZHAI, 2012). Sendo assim, é importante certificar-se de que os dados textuais tenham sido corretamente estruturados e que palavras irrelevantes (*stopwords*) tenham sido retiradas, tanto como textos que não possuem nenhum significado para a análise.

Outra fase de grande importância no *Text Mining* é a mineração da informação. Segundo Hoeschl *et al.* (2002), o *Text Mining* possui duas fases principais e subsequentes que são a de extração da informação e a de mineração de informação. São consideradas principais, pois uma destina-se a extrair conceitos relevantes para a análise e estruturá-los, e a outra a aplicar as técnicas adequadas de análise para obter bons resultados.

Primeiramente deve-se decidir a estrutura dos dados que serão coletados. Para Feldman (1995), dada as tecnologias de processamento robusto de textos que existem, precisa-se definir a estrutura dos dados levando em conta a relevância da informação e o custo. Sendo assim, é preciso pensar em uma estrutura que atenderá a demanda de pesquisa, gastando o menos possível.

Tendo as informações estruturadas, é necessário definir quais palavras serão levadas em consideração para a extração de conteúdo. Essa extração de informação tem uma variedade de domínios, então o tipo e estrutura da informação extraída dependem da necessidade da aplicação. Por exemplo, pesquisadores de biomedicina precisam de dados de publicações científicas, enquanto profissionais financeiros estão interessados em artigos de jornais econômicos para ajudá-los em tomadas de decisão (GRUPTA; LEHAL, 2009).

Um exemplo de formato para extração de dados pode ser dado por uma ficha policial que busca um determinado tipo de gangue foragida. Eles têm como identificar supostas ameaças por algumas informações que são compatíveis com a gangue, como o carro utilizado e o tipo de roubo. A Figura 1 mostra o exemplo de um texto e de como seria esse quadro com as informações extraídas.

Polícia tenta achar gangue de roubo de postos de combustível na cidade de Santa Maria com carro Corsa Prata (ZH NOTÍCIAS, 2016).

Quadro de Roubos	
Ano	2016
Local	Posto de Combustível
Cidade	Santa Maria
Carro	Corsa Prata

Figura 1: Quadro indicativo de roubos.

2.1 Técnicas de *Text Mining*

Segundo Wives (2002), os tipos de técnicas de *Text Mining* são: sumarização, classificação/categorização e *clustering*. Hoje a literatura mostra que além dessas três técnicas

existem ainda outras: extração de informação, *Topic Tracking*, *Concept Linkage*, Informação Visual, *Question Answering* e *Association Rule Mining* (GRUPTA; LEHAL, 2009).

2.1.1 Sumarização

Esta técnica consiste em reduzir o tamanho e detalhes de um documento, mantendo seus principais pontos e seu significado geral. Sendo assim, esse processo irá gerar sentenças com as palavras mais importantes do texto, o que torna possível a compreensão do texto sem necessidade de leitura (GRUPTA; LEHAL, 2009).

Quando pessoas resumem textos é necessária a leitura total do texto primeiramente para depois escreverem um resumo com o que acham relevantes. Muitas vezes as pessoas levam em consideração palavras e fatos que são irrelevantes para a compreensão do texto, enquanto o processo de sumarização tende a escolher somente os pontos mais importantes e que possibilitem o entendimento geral do documento.

2.1.2 Classificação / Categorização

O método de Classificação identifica a classe que o documento pertence, tendo um conjunto de características pré-definidas para cada classe. A Categorização identifica os temas principais de um documento já tendo uma lista de tópicos anteriormente definida (GRUPTA; LEHAL, 2009).

Segundo os autores, ao categorizar um documento, o computador trata esse documento como um "saco de palavras". Sendo assim, o programa conta as palavras que aparecem no texto e a partir disso identifica os tópicos principais. Normalmente, ferramentas de categorização também contam com um método de *ranking* que diz a ordem dos documentos com mais similaridade para cada tópico.

2.1.3 *Clustering*

Clustering é uma técnica para encontrar grupos de textos similares nos documentos de acordo com suas características, sendo diferente da Categorização, pois não utiliza tópicos pré-definidos para fazer o agrupamento.

Levando em consideração que documentos podem ter uma infinidade de tópicos e que muitas vezes é impossível definir esses tópicos anteriormente, como é feito na Categorização, a técnica de *Clustering* faz com que não se perca informações, pois leva em consideração todos os tópicos que possam vir a aparecer. Portanto, toda informação que venha a ser relevante não é perdida.

2.1.4 Extração de Informação

O *software* de extração da informação identifica frases chaves e relações dentro do texto. Isto é feito procurando sequências pré-definidas no texto, em um processo chamado de padrão de correspondência. Esta tecnologia pode ser muito útil ao lidar com grandes volumes de textos (GRUPTA; LEHAL, 2009).

2.1.5 *Topic Tracking*

Topic Tracking consiste basicamente em prever documentos de interesse para as pessoas a partir do que elas veem ou compram, seja em sites de compras, busca, entre outros. Um exemplo pode ser dado pelos e-mails recebidos quando se compra no site da *Amazon* e depois a loja manda uma lista de artigos relacionados para futuras compras.

2.1.6 *Concept Linkage*

Concept Linkage é um método que tende a conectar documentos relacionados utilizando seus conceitos, o que faz com que as pessoas, muitas vezes, encontrem mais informações do que em uma pesquisa tradicional.

É um conceito muito valioso no *Text Mining* segundo Grupta e Lehal (2009), especialmente nas áreas biomédicas, em que é quase impossível os pesquisadores lerem todo material disponível para determinada pesquisa e fazerem associações com outras pesquisas. Idealmente, *Concept Linkage* pode identificar ligações entre doenças e tratamentos, por exemplo, que uma pessoa não poderia.

2.1.7 Informação Visual

Também chamado de *Visual Text Mining*, essa técnica tende a colocar grandes bases textuais em um mapa fornecendo navegação e busca. *Visual Text Mining* é útil quando o usuário

precisa examinar muitos textos ou documentos e ainda fazer uma associação de tópicos entre eles.

O governo pode usar a Informação Visual, por exemplo, para identificar redes terroristas ou para encontrar informações sobre crimes organizados. Esta técnica pode fornecer um mapa com as possíveis relações entre as atividades suspeitas para uma investigação (GRUPTA; LEHAL, 2009).

2.1.8 *Question Answering* (Q&A)

Question Answering é o método para encontrar a melhor resposta para uma questão dada. Muitas vezes esse método usa mais de uma técnica de *Text Mining*. Um exemplo dessa técnica é a lista de "perguntas frequentes" que é encontrada em muitos sites, onde existem as perguntas que são comuns de serem feitas e suas melhores respostas. Isso evita, muitas vezes, que a pessoa que está utilizando o site entre em contato por telefone, e-mail ou outro meio para sanar alguma dúvida já resolvida. Além disso, ajuda a pessoa e a empresa a ganharem tempo.

2.1.9 *Association Rule Mining (ARM)*

Association Rule Mining é uma técnica usada para descobrir as relações entre uma grande quantidade de variáveis em um conjunto de dados. Descobre relações em um conjunto grande de dados contendo duas ou mais variáveis, sendo similar a análise de correlação (GRUPTA; LEHAL, 2009).

Segundo os autores, *ARM* vem sendo muito usada nos processos de tomada de decisão das empresas. *ARM*, por exemplo, pode descobrir que itens os consumidores tendem a comprar juntos e, portanto os supermercados podem colocar esses produtos pertos para aumentar as vendas. Outro exemplo pode ser visto em notícias, levando em conta quando aparece determinado conjunto de palavras como "inflação" e "dinheiro" é altamente provável que o mercado de ações também é mencionado.

2.2 *Text Mining* e suas aplicações

Nesta seção serão apresentados alguns exemplos onde o *Text Mining* é utilizado. Assim será possível ver o quão útil o *Text Mining* pode ser e que áreas pode abranger, além de

demonstrar o quanto o uso dos textos pode ser útil nas análises estatísticas. Outros exemplos interessantes são apresentados por Silva (2013).

2.2.1 Análise Competitiva na Mídia Social na Indústria de Pizzas

Esse estudo de caso faz uma análise dos dados das mídias sociais *Facebook* e *Twitter* das três maiores franquias de pizzas: *Pizza Hut, Domino's Pizza e Papa John's Pizza*. O objetivo é utilizar esse grande volume de dados para melhorar as companhias de acordo com as opiniões dos clientes, além de desenvolver uma estratégia de análise nas mídias sociais. Esse estudo foi divulgado no artigo "*Social media competitive analysis and text mining: A case study in the pizza industry*" (HE; ZHA; LI, 2012).

A Figura 2 mostra o número de *tweets* por dia no mês de Outubro de 2011.

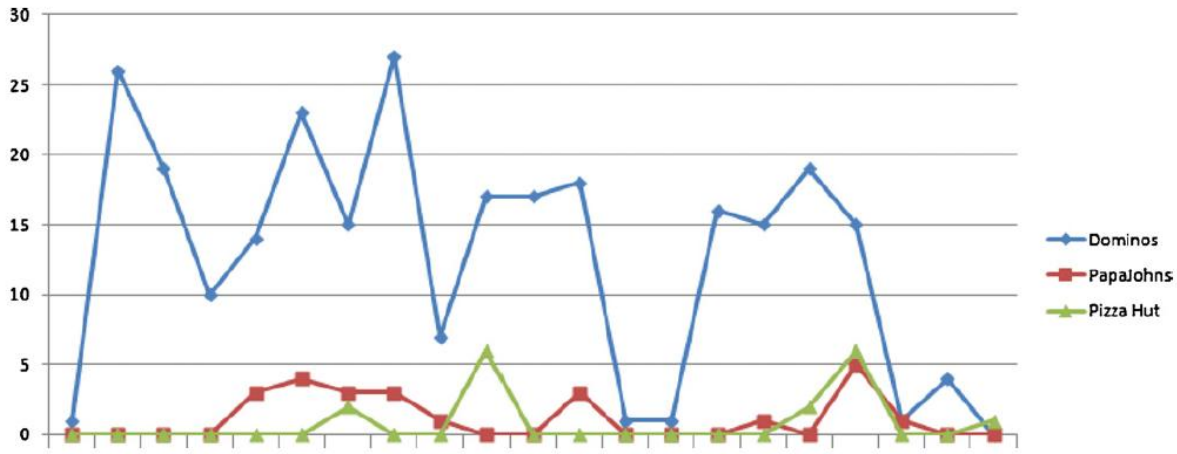

Figura 2: Número de tweets por dia no mês de Outubro das três companhias de pizza.

Para cada companhia de pizza foram encontrados cinco temas principais no *Twitter*: pedido e entrega, qualidade da pizza, retorno dos clientes para decisão de compra, *tweets* sociais e *marketing tweets.* Enquanto no *Facebook* foram encontrados seis temas principais e postagens de fotos e de agradecimentos.

Sendo assim, as companhias de pizzas podem ver através dos tópicos encontrados sobre a qualidade de seu serviço e como e onde devem melhorar. As mídias sociais são um bom local para se achar informações sobre diversos negócios.

2.2.2 *Twitter* e a Copa do Mundo do Brasil

Um estudo foi feito com *tweets* postados antes do início da Copa do Brasil de 2014 para achar os diferentes tópicos discutidos sobre o assunto no *Twitter*. O estudo foi divulgado no artigo "*A Case Study in Text Mining: Interpreting Twitter Data From World Cup Tweets*" (GODFREY *et al.*, 2014).

Foram utilizadas análises de *cluster* usando *k-means* e *Non-Negative Matrix Factorization* (NMF) para comparar os resultados. Um dos problemas enfrentados antes da análise foram os ruídos, que nesse caso são *tweets* não relevantes.

Gráficos feitos no *Software Gephi* mostram os diferentes tópicos encontrados usando os dois tipos de análises citadas. As Figuras 3 e 4 mostram os gráficos para *k-means* e *NMF*, respectivamente.

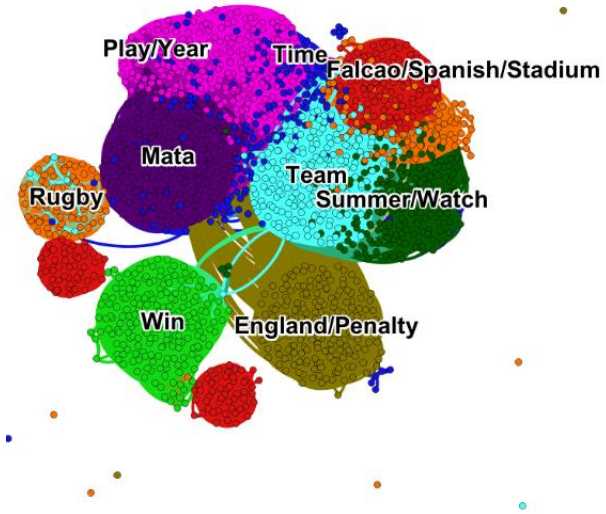

Figura 3: Tópicos por *k-means*

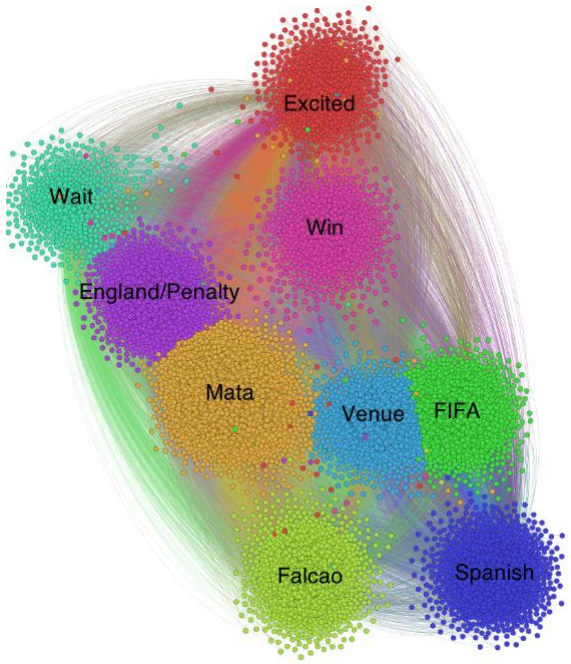

Figura 4: Tópicos por *NMF*

No gráfico para *k-means*, mostrado na Figura 3, é possível ver quanto os diferentes tópicos dos *tweets* estão agrupados, onde os *tweets* agrupados com maior frequência são representados por uma cor diferente no gráfico com seu respectivo tópico nomeado. Pode haver mais de um tópico dividido entre os *tweets*, como por exemplo, 'Falcao/*Spanish*/*Stadium*'.

O gráfico para *NMF*, mostrado na Figura 4, segue a mesma ideia, representando pelos nós coloridos os diferentes tópicos nos quais os *tweets* são relacionados, além de que quanto mais próximo um tópico está do outro, maior a relação entre eles, como, por exemplo, '*Venue'* e '*Fifa'* são tópicos que tem relação, pois estão próximos.

Os dois tipos de análises encontraram uma coleção de tópicos parecidos nos *tweets*, mas a técnica *NMF* foi mais rápida para obter resultados e mais fácil de interpretar do que a técnica de *k-means*.

2.2.3 *Text Mining* na Biomedicina

Um artigo foi publicado sobre como usar os diferentes métodos de *Text Mining* a partir de cinco passos para projetos de trabalhos em Biomedicina. O Artigo foi intitulado "*Five Steps to Text Mining in Biomedical Literature*" (MATHIAK; ECKSTEIN, 2004).

Os procedimentos do *Text Mining* foram divididos em cinco passos para poder aperfeiçoar cada etapa individualmente. Os cinco passos foram: coleta de texto, pré-processamento de texto, análise dos dados, visualização e validação.

Um teste foi feito com um conjunto de textos do "*PubMed Central Open Access Initiative*" (PMC OAI), utilizando *Clustering* para conectar os documentos similares. A Figura 5 mostra um gráfico desses documentos e sua aproximação quanto à similaridade (MATHIAK; ECKSTEIN, 2004).

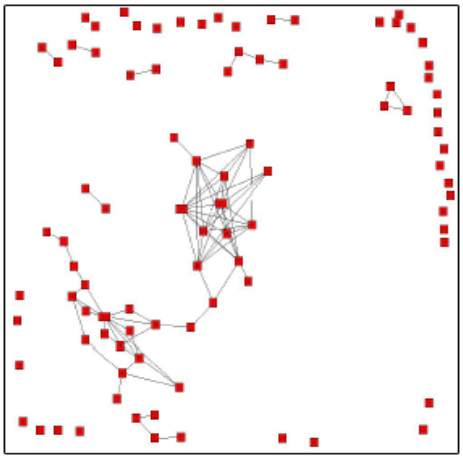

Figura 5: Gráfico de Similaridade dos documentos do PMC OAI.

Segundo os autores, o *Text Mining* vem se mostrando efetivo na área da Biomedicina e cresce cada vez mais seu uso para projetos de trabalhos de conclusão na Graduação e Pós-Graduação dessa área.

2.2.4 *Text Mining* no R

Além dos diversos pacotes e técnicas disponíveis no *Software* R, a análise de *Text Mining* possui ainda funções como a capacidade de mostrar em mapa mundial a quantidade de seguidores que usaram determinada *hashtag* (#, usada para falar de determinado assunto com destaque). Também é possível demonstrar em gráfico os usuários mais influentes para determinado estudo de *tweets*, ou seja, as pessoas que mais escreveram sobre determinado assunto, e também se foram copiadas (*retweets*). Gráficos de linhas com o máximo número de *tweets retweetados* (*tweets* repetidos de alguém que escreveu antes) por tempo, seja por dia, mês ou ano também são feitos na análise de *Text Mining* atuais. Esses exemplos são para análises feitas com dados do *Twitter*. A Figura 6 mostra um exemplo de gráfico que mostra, com pontos

verdes em tom mais claro, os locais de onde as pessoas *tweetaram* sobre @RDataMining (ZHAO, 2016).

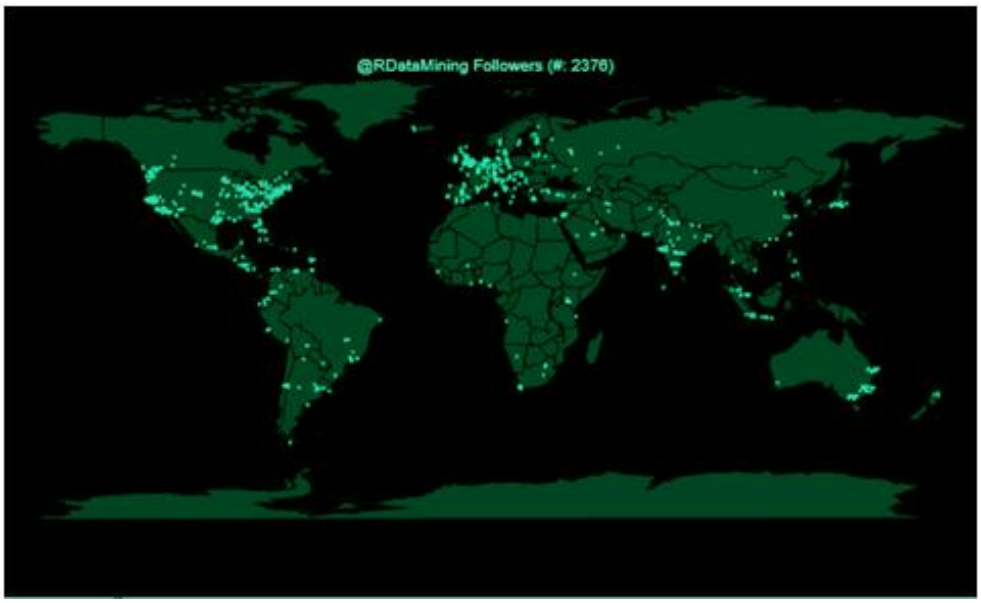

Figura 6: Mapa de pessoas que tweetaram sobre @RDataMining (ZHAO, 2016).

3. MATERIAL E MÉTODO

Nesta seção será apresentado o *software* utilizado, mostrando também como os dados foram coletados e que técnicas do *Text Mining* foram utilizadas na análise.

3.1 *Software* R

O *Software* usado para esse trabalho foi o *R* versão do *R* 3.2.4, sendo escolhido por ser um *software* que contém recursos para análise de *Text Mining*, por ser gratuito e por já ter sido utilizado em análises de circulação de livros anteriores do programa *One Book One Chicago*. O *Software R* pode ser baixado diretamente na internet pelo site 'https://www.r-project.org/'. Além disso, o *Software R* tem amplo conjunto de funções e pode ser aperfeiçoado com o uso de novos pacotes, ou seja, é um programa bastante poderoso quanto a análises estatísticas.

Também é possível acessar o *Software R* remotamente por um *Server* que pode ser acessado pela internet, tendo assim cada conjunto de usuários uma máquina virtual protegida por senha. Cada usuário terá seu *login* e senha que dará acesso a máquina e também limitará suas

operações que são controladas por um administrador. Isso facilita muito a interação entre as pessoas e troca de informações, de forma que todos podem ver e ajudar seus colegas nas análises. O programa *One Book One Chicago* conta com uma máquina virtual poderosa onde são armazenadas suas análises, isso tudo para ter um trabalho seguro e ao mesmo tempo compartilhado com os participantes.

O pacote '*tm'* é o pacote do R que possui as principais funções utilizadas no *Text Mining*, sendo o mais utilizado nas análises. Alguns pacotes que serão utilizados nesse trabalho: '*RColorBrewer*', '*fpc*', '*wordcloud', 'topicmodels' e 'ggplot2'* que servem para colocar cores em mapas temáticos, auxílio na análise de *cluster*, nuvem de palavras, modelagem por tópicos e construção de gráficos, respectivamente.

3.2 Algumas novidades de *Text Mining* no R

Alguns pacotes para *Text Mining* foram agregados ao R nos últimos tempos. Os métodos e novos pacotes que serão utilizados no trabalho estão a seguir:

- Pacote '*stringr'* – pacote que ajuda com o manuseio de *strings*. Utilização nesse trabalho para excluir *tweets* que continham palavras indesejáveis.

- Pacote '*RColorBrewer'* – este pacote serve para colocar cores para mapas temáticos. Nesse trabalho será usado para destacar as palavras na nuvem de palavras.

- Pacote '*fpc'* – pacote para auxílio de métodos, validação e estimação de *clusters*. Será usado nesse trabalho para fazer análise de *Cluster around medoids*.

- Pacote '*topicmodels'* – pacote utilizado para modelagem por tópicos, que é o processo de dividir os textos por tópicos. É possível também fazer gráficos com cores dividindo os diferentes tópicos.

- Pacote '*cluster'* – pacote que contém funções para análises de *cluster*. Utilização nesse trabalho para fazer gráfico '*cusplot'*.

Além desses pacotes, o processo *Wordclouds* por *cluster*, que proporciona melhor visualização e entendimento das informações textuais, é também novidade para a análise de *Text Mining* no R. Serão feitas nuvens de palavras para cada *cluster* encontrado na análise de *k-means* e para cada tópico encontrado na modelagem por tópicos.

3.3 Dados do *Twitter*

Os dados desse trabalho foram coletados da rede social *Twitter*, que permite aos usuários receberem e enviarem textos com até 140 caracteres. Os textos provenientes dessa rede social são chamados de *tweets*. O *Twitter* é bastante usado mundialmente sendo apresentado em 37 idiomas.

Foram recebidos 3606 *tweets* distribuídos em 10 arquivos '.csv' que foram previamente obtidos do *Twitter* por um participante do programa *One Book One Chicago*, sendo coletados em diferentes horários. A coleta foi feita a partir de palavras que se relacionam com o programa *OBOC*. O livro que estava sendo divulgado era o *Third Coas*t do autor Thomas Dyja que conta a história da cidade de Chicago pós Segunda Guerra Mundial.

Os dados foram carregados no *R* através da função *read.csv* com codificação para strings (variáveis de texto), como apresentado a seguir.

```
twitter1 <- read.csv("tw_20150907_1600.csv",TRUE,",", encoding="UTF-8", stringsAsFactors=FALSE)
twitter2 <- read.csv("tw_20150908_1430.csv",TRUE,",", encoding="UTF-8", stringsAsFactors=FALSE)
twitter3 <- read.csv("tw_20150909_0630.csv",TRUE,",", encoding="UTF-8", stringsAsFactors=FALSE)
twitter4 <- read.csv("tw_20150910_0930.csv",TRUE,",", encoding="UTF-8", stringsAsFactors=FALSE)
twitter5 <- read.csv("tw_20150910_2230.csv",TRUE,",", encoding="UTF-8", stringsAsFactors=FALSE)
twitter6 <- read.csv("tw_20150910_2359.csv",TRUE,",", encoding="UTF-8", stringsAsFactors=FALSE)
twitter7 <- read.csv("tw_20150913_2030.csv",TRUE,",", encoding="UTF-8", stringsAsFactors=FALSE)
twitter8 <- read.csv("tw_20150915_0830.csv",TRUE,",", encoding="UTF-8", stringsAsFactors=FALSE)
twitter9 <- read.csv("tw_20150915_1830.csv",TRUE,",", encoding="UTF-8", stringsAsFactors=FALSE)
twitter10<- read.csv("tw_20150917_1430.csv",TRUE,",", encoding="UTF-8", stringsAsFactors=FALSE)
```

Após os dados serem carregados, foram compilados em um objeto chamado '*twitterall'*, usando a função de combinar linhas que é a '*rbind'*.

```
twitterall =rbind(twitter1,twitter2,twitter3,twitter4,twitter5,twitter6,twitter7,twitter8,twitter9,twitter10)
```

3.4 Etapas do *Text Mining*

Segundo Dixon (1997) e Morais e Ambrósio (2007), são quatro as etapas do *Text Mining*: 1) Identificação do problema, 2) Pré-processamento, 3) Mineração dos dados e 4) Pós-processamento. A Figura 7 mostra essas etapas.

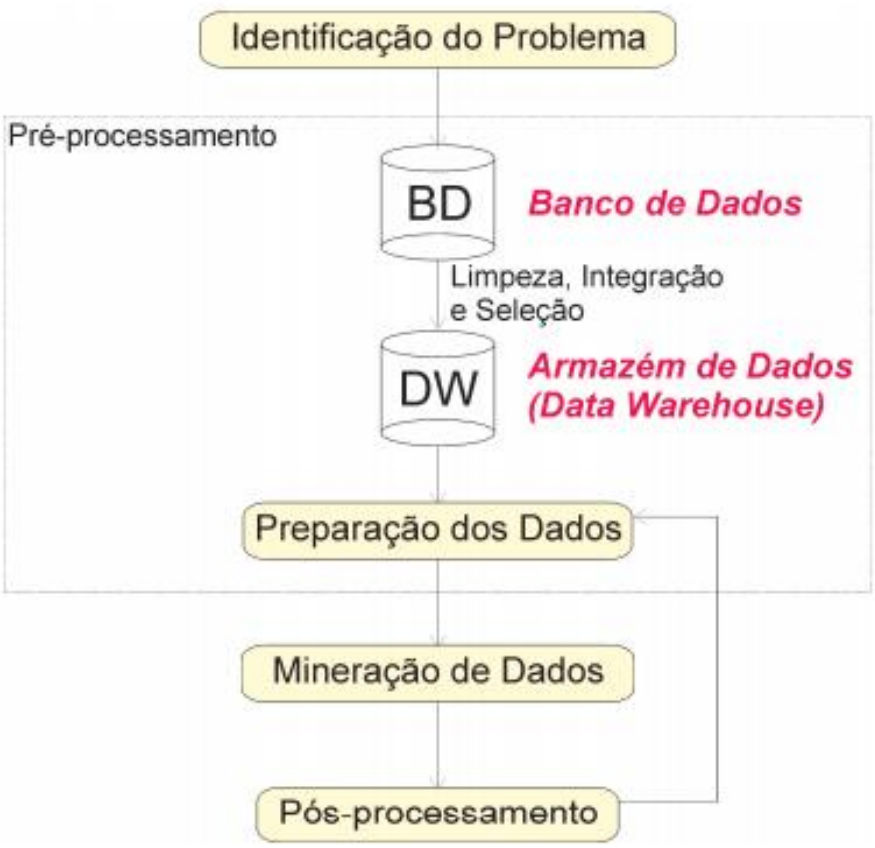

Figura 7: Etapas do processo de *Text Mining* (MORAIS; AMBRÓSIO, 2007).

Cada uma destas etapas será aplicada no estudo de caso proposto e terão seus resultados apresentados na seção 4 desse trabalho.

1. Identificação do Problema: pesquisadores do programa *OBOC* gostariam de saber o que estava sendo falado sobre o programa na rede social *Twitter*.
2. Pré- Processamento
   a. Limpeza, integração e seleção dos dados: dados do *Twitter* foram carregados no *R*, a coluna dos *tweets* foi separada e os *tweets* que não eram sobre o programa *OBOC* foram retirados, assim também como os *tweets* duplicados.
   b. Armazenagem dos dados: a coluna que contém os textos dos *tweets* formou o chamado *corpus*, que nada mais é que a coleção de textos.
   c. Preparação dos dados: remover dados desnecessários do *corpus*, tais como pontuação, números, *urls* e *stopwords*.

3. Mineração dos dados: uma matriz de associação é criada para a identificação de frequência dos termos e associação entre eles. Gráfico de linhas e nuvem de palavras são feitos para mostrar termos mais frequentes. As técnicas de *Clustering* são utilizadas para agrupar os *tweets* por diferentes assuntos, assim como Modelagem por Tópicos.
4. Pós- Processamento: conclusões sobre as análises feitas e interpretações sobre a resolução do problema proposto pelo programa *OBOC*.

4. RESULTADOS

Os resultados das etapas do *Text Mining* utilizadas na análise dos dados da rede social *Twitter* sobre o programa *One Book One Chicago - OBOC* serão apresentados nessa seção. A obtenção dos dados e como foram carregados no *R* foram mostrados na seção 3, portanto o ponto de partida será a limpeza dos dados.

4.1 Limpeza, integração e seleção dos dados

O primeiro passo foi tirar os *tweets* duplicados, pois poderia haver *tweets* iguais, em função da forma como foi feita a coleta. Um exemplo seria que a coleta feita em partes poderia ter começado às dez horas da manhã de um dia e terminado às duas horas da tarde e outra coleta sendo feita no mesmo dia poderia começar à uma hora da tarde e terminar às cinco horas da tarde. Para a retirada desses valores duplicados foi usada uma função de exclusão que levava em consideração a identidade de cada *tweet*, partindo do pressuposto que cada *tweet* tem seu próprio número de identidade, ou seja, quando alguém escreve algo no *Twitter* esse texto possui um número de identificação próprio. O banco de dados foi ordenado pelo número de identificação '*id_str*' (banco1) e depois uma função de exclusão auxiliar identificou valores iguais e assim um novo banco (banco2) sem valores duplicados foi criado. O banco com valores duplicados tinha 3606 *tweets* e o novo banco passou a ter 3134 *tweets*.

```
banco1 <- twitterall[ order(twitterall$id_str),]
exclusao <- which(duplicated(banco1)==T) # cria vetor com duplicados
aux <- exclusao - 1
exclusao1 <- unique(c(exclusao, aux))
banco2 <- banco1[-exclusao1, ]
```

Como algumas colunas que contém informações, como, por exemplo, a do nome de usuário, não serão utilizadas, foi criado um objeto chamado de '*banktext'* que contém somente a coluna dos *tweets*.

```
banktext = banco2[,3]
```

Após essa limpeza inicial, ainda havia *tweets* que não tinham relação com o programa, pois a busca feita foi por palavras e acarretou em textos que tinham a ver com outros eventos ou assuntos semelhantes. Um exemplo disso foi que na mesma época da divulgação do livro havia um evento de música na cidade de Chicago chamado *'Third Coast'*, que é o mesmo nome do livro divulgado em 2015, portanto *tweets* desse evento também estavam presentes. Para a exclusão desses *tweets* indesejáveis foi utilizada a função '*lapply'* e '*unique'* do pacote '*stringr'* do R, usadas, respectivamente, para identificarem onde se encontram um conjunto de palavras previamente escolhido e remover as linhas que contém essas palavras. As palavras escolhidas primeiramente foram aquelas que ao olhar superficialmente nos arquivos, foi possível identificar que não estavam relacionadas ao *OBOC*, conforme os comandos que seguem.

```
library(stringr)
words <- c("Hannibal", "beer", "dead", "radio", "Drinking", "Race", "Gold","gold", "Cancer","cancer"
,"Gulf","Edge", "Hip")
pos <- lapply(X = words, FUN = function(w) grep(pattern = w, x = banktext))
pos2 <- unique(do.call(what = c, args = pos))
banktext[pos2] #para mostrar todos os tweets que foram retirados
banktext <- banktext[-pos2]
```

Excluídos os primeiros *tweets* ainda havia informação não condizente com o programa, portanto foi utilizado um conjunto de palavras que com certeza estariam presentes nos *tweets* sobre o programa *OBOC*, onde as palavras foram escolhidas também a partir de uma leitura superficial. Foi utilizada essa abordagem, pois o processo contrário dificultou a visualização e exclusão dos textos irrelevantes, já que usando as palavras não relacionadas ao programa, ainda sobravam muitos *tweets* que não eram de interesse. Tendo isso em vista, o novo '*banktext'*, utilizando os comandos listados a seguir, passou a ter 674 *tweets*.

```
words <- c("BOOK", "book", "Book", "Thomas", "author","OBOC" ,"reading")
pos <- lapply(X = words, FUN = function(w) grep(pattern = w, x = banktext))
pos2 <- unique(do.call(what = c, args = pos))
banktext = banktext[pos2]
```

4.2 Armazenagem dos dados

A parte textual foi extraída a partir do comando '*Corpus'* para a construção da coleção de textos que nesse trabalho foi chamado de '*mycorpus'*. Anteriormente foi necessária a conversão em dois passos dos dados textuais para o padrão internacional, nos quais o novo objeto com a coluna de textos passou a ser chamado '*banktextconv2'*. A partir dessa parte do trabalho o pacote de *Text Mining* '*tm'* será utilizado.

```
library(tm)
banktextconv <- enc2utf8(banktext)
banktextconv2 <- iconv(banktextconv, to="ASCII//TRANLIT", sub="byte")
mycorpus = Corpus(VectorSource(banktextconv2))
```

4.3 Preparação dos dados

Com o *corpus* formado, o próximo passo é preparar os dados, excluindo palavras sem significado, antes de as análises serem feitas. As letras foram transformadas todas para minúsculas, foram retiradas as pontuações, os números e as *url`s* (informação padronizada de documentos da internet).

```
mycorpus = tm_map(mycorpus, content_transformer(tolower))
mycorpus = tm_map(mycorpus, removePunctuation)
mycorpus = tm_map(mycorpus, removeNumbers)
removeURL = function(x) gsub("http[[:alnum:]]*","",x)
mycorpus = tm_map(mycorpus, removeURL)
```

Foi criada uma cópia do *corpus* para contar a frequência das palavras e exemplificar os *tweets*, pois uma função para codificar o *corpus* foi usada e não era possível fazer esses passos posteriormente.

```
mycorpuscopy = mycorpus
mycorpus = tm_map(mycorpus, PlainTextDocument)
```

Com os comandos aplicados, os *tweets* ficaram como mostra o exemplo após os comandos de inspeção.

```
for (i in 5:10){
  cat(paste("[[", i, "]]", sep = ""))
  writeLines(mycorpuscopy[[i]])
}
```

```
[[2]]british film institute film  television handbook   eddie dyja pb book
[[3]]rt bookchicago  know  know  really wanna know   new oboc   coming soon   wait  past picks
[[4]] day long tour  presentations inspired  oboc book  stationeleven
```

Após serem excluídas pontuações, números e *url's,* as *stopwords* foram as próximas a serem retiradas. *Stopwords* são palavras que não contém significado relevante para fins de conclusões de análises, como, por exemplo, artigos, preposições, conjunções e pequenas palavras específicas.

```
mycorpus = tm_map(mycorpus, removeWords, stopwords("english"))
mycorpuscopy = tm_map(mycorpuscopy, removeWords, stopwords("english"))
```

Sendo os *tweets* todos escritos em inglês não foi necessária a implantação de uma nova lista de *stopwords*, pois a lista que o *Software R* contém é bem vasta na língua inglesa. Caso os *tweets* fossem em português, por exemplo, existem alguns sites que contêm listas de *stopwords* que podem ser utilizadas na análise de *Text Mining*.

4.4 Mineração dos dados

Nesta subseção serão apresentados os passos e resultados da análise de *Text Mining* feita após a preparação dos dados.

Uma matriz de associação é criada para a identificação de frequência dos termos e associação entre eles. Gráfico de linhas e nuvem de palavras foram construídos para mostrar termos mais frequentes. A técnica de *Clustering* foi utilizada para agrupar os *tweets* em grupos por diferentes assuntos, mostrando também o dendograma de palavras para saber quantos grupos seriam formados.

4.4.1 Frequência de Palavras

Algumas palavras que pressupostamente são as mais apresentadas no texto e podem aparecer com grande frequência foram contadas. Essas palavras foram: "*coast"*, "*third"*, "*OBOC"* e "*book"*. As frequências estão descritas abaixo, respectivamente, após de ser mostrado o comando para contar as frequências.

```
coastCases = tm_map(mycorpuscopy, grep, pattern = "\\<coast")
sum(unlist(coastCases))
thirdCases = tm_map(mycorpuscopy, grep, pattern = "\\<third")
sum(unlist(thirdCases))
obocCases = tm_map(mycorpuscopy, grep, pattern = "\\<oboc")
```

```
sum(unlist(obocCases))
bookCases = tm_map(mycorpuscopy, grep, pattern = "\\<book")
sum(unlist(bookCases))
```

```
[1] 516
[1] 512
[1] 151
[1] 608
```

Observa-se, portanto, que a palavra "*coast"* foi citada 516 vezes, a palavra "*third"* 512 vezes, a abreviatura do programa de leitura da cidade de Chicago, "*oboc"*, 151 vezes e a palavra "*book"* 608 vezes.

4.4.2 Matriz de Termos

O conjunto de textos é transformado em uma matriz de termos, contendo as palavras em cada linha e os documentos (cada *tweet*) em cada coluna. Sendo assim, obtém-se quantas vezes cada palavra apareceu em cada documento. O comando para essa matriz está a seguir.

```
tdm = TermDocumentMatrix(mycorpus, control = list(wordLengths = c(3, Inf)))
```

Como exemplo, foi inspecionado os primeiros 5 *tweets* para ver se continham as palavras "*abc"*, "*abcchicago"* e "*absurdly"*, sendo possível perceber, na Figura 8, que a palavra "*absurdly"* aparece uma vez no primeiro *tweet*.

```
inspect(tdm[0:5,1:5])
```

	Tweets				
Palavras	1	2	3	4	5
abc	0	0	0	0	0
abcchicago	0	0	0	0	0
absurdly	1	0	0	0	0

Figura 8: Matriz de frequência de termos

4.4.3 Palavras mais frequentes

Tendo a matriz de termos '*tdm'*, é possível inspecionar as palavras mais frequentes. Usando uma frequência mínima de 100 (*lowfreq*=100) pode-se usar a função '*findFreqTerms'* para mostrar as palavras (também chamadas de termos) que aparecem nessa condição. Na Figura 9 são apresentadas as palavras mais frequentes, utilizando o seguinte comando.

```
(freq.terms = findFreqTerms(tdm, lowfreq = 100))
```

book	bookchicago	chicago
coast	next	oboc
one	third	thomas
title	wednesday	

Figura 9: Palavras com frequência maior que 100

Com essa informação de palavras mais frequentes é possível saber os assuntos que aparecem nos *tweets.* Além de saber o assunto, é uma forma de saber se a análise está seguindo o caminho correto, pois mostra as palavras que seriam supostamente esperadas. Caso aparecessem palavras que não tenham nenhum sentido com o programa que está sendo trabalhado, isso seria sinal de que algo estaria errado na análise. Nesse caso temos palavras condizentes com o programa *One Book One Chicago*, o que indica que a análise está seguindo o caminho correto.

Para observar a frequência dos termos de modo mais visual, um gráfico de barras foi feito, mostrando os termos com suas frequências (Figura 10). Os comandos que geram esse gráfico com palavras de frequência com valores mínimos de 100 nos *tweets* coletados sobre o programa *OBOC* estão apresentados a seguir. Primeiramente é usada uma função '*rowSums'* que soma quantas vezes cada palavra aparece na matriz de termos. Seguindo, um objeto novo chamado '*df*'' é formado a partir da função '*data.frame'* para ser criada uma tabela com duas colunas, a primeira contendo o nome de cada palavra e a segunda contendo as frequências associadas a essas palavras. O gráfico foi criado usando a função '*ggplot'* que está contida no pacote '*ggplot2'*.

```
term.freq = rowSums(as.matrix(tdm))
term.freq = subset(term.freq, term.freq >=100)
df = data.frame(term = names(term.freq), freq = term.freq)
library(ggplot2)
ggplot(df, aes(x = term, y = freq))+geom_bar(stat = "identity", fill="blue", colour = "black")+
  xlab("Termos")+ ylab("Freq.") + coord_flip()
```

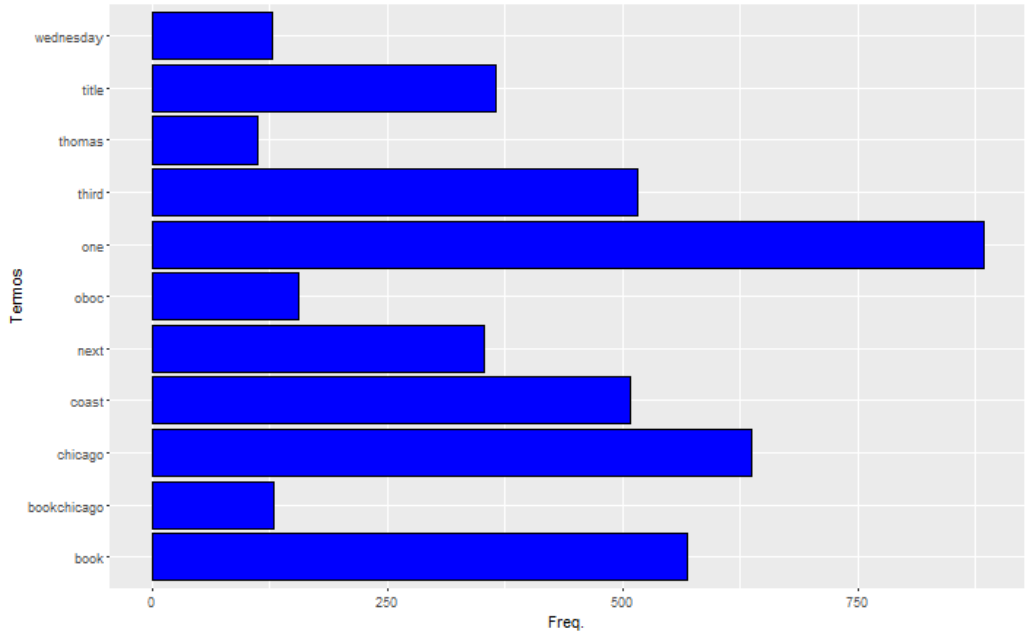

Figura 10: Gráfico de termos e suas frequências

4.4.4 Associação entre palavras

É possível achar quais palavras estão associadas com outras, por meio da função '*findAssocs'* disponível no pacote '*tm'*. Essa função tem o objetivo de calcular através da matriz de termos a correlação entre as palavras dos textos que estão sendo usados na análise, sendo possível determinar a correlação mínima que o usuário quer utilizar. Essa e todas as outras funções do pacote '*tm'* podem ser encontradas com explicações e exemplos no arquivo do pacote com o nome "*Introduction to the tm Package Text Mining in R*" (FEINERER, 2015), fornecido pelo próprio *Software R* em seu site (https://cran.r-project.org/web/packages/tm/tm.pdf).

Os primeiros termos que se procuraram associações foram os que tinham maior frequência nos *tweets*. Na função '*findAssocs'* é necessário indicar o termo que se deseja obter as associações entre aspas duplas e o valor mínimo de correlação que será aceito. Nesse primeiro exemplo foi usada a palavra "*one"* que quer dizer "um" em inglês, presente no nome do programa, com associação de valor mínimo de 0.3 como na função descrita a seguir e seus resultados de valores de associação de acordo com cada palavra.

```
findAssocs(tdm, "one", 0.3)
$one
   title  chicago     next     book    third    coast wednesday
    0.78     0.75     0.73     0.71     0.48     0.47      0.35
```

A palavra “*one”* mostrou sete palavras que são as mais associadas a ela. As palavras em inglês “*title*”, ”*chicago*”, ”*next*”, ”*book*”, ”*third*”, ”*coast*” e ”*wednesday*” são traduzidas para português como “título”, ”Chicago”, ”próximo”, ”livro”, ”Terceira”, ”Costa” e ”quarta-feira”, respectivamente. O termo mais correlacionado com a palavra “*one*” foi “*title*” com valor de 0.78 que é um valor bastante alto de correlação. Sendo assim, é possível ver com essas palavras relacionadas que o próximo livro que será lançado para o programa *One Book One Chicago* será o *Third Coast* e isso provavelmente será feito na quarta-feira.

Seguindo para a próxima palavra mais frequente, “*Chicago*”, utilizou-se agora o mínimo de correlação de 0.15.

```
findAssocs(tdm, "chicago", 0.15)
$chicago
      one      next     title   morning    public     third
     0.75      0.67      0.64      0.53      0.53      0.50
    coast      book wednesday    system   tribune   library
     0.49      0.45      0.45      0.44      0.44      0.29
 american     built     dream       lib socialinchicago      news
     0.21      0.21      0.21      0.20      0.17      0.16
```

Nesse caso, pode-se ver que o próximo título para o programa *One Book One Chicago* da biblioteca pública da Cidade será o *Third Coast*, sendo noticiado na quarta-feira pela manhã no jornal chamado *Chicago Tribune*. Além disso, é possível perceber que é falado o tema do livro, que é construir o sonho americano, pois, como já foi dito, o livro trata da reconstrução da cidade de Chicago pós Segunda Guerra Mundial. As palavras que fazem com que se tire essa conclusões são em inglês “*one*”, ”*book*”, ”*next*”, ”*title”*, ”*public*” (pública), ”*library*” (biblioteca), “*morning*” (manhã), “*Wednesday*”, “*news*” (notícias), “*tribune*” (Tribuna), “*american*” (americano), “*dream*” (sonho),”*built*” (construído), “*third*”, “*coast*”.

Como as outras palavras que apareciam com mais frequência já apareceram associadas às palavras “O*ne*” e “*Chicago*”, foram buscadas novas palavras com frequências um pouco menores, sendo uma delas o primeiro nome do autor do livro (Thomas), correlação mínima 0.3.

```
findAssocs(tdm, "thomas", 0.3)
$thomas
    dyja beginning remember chipublib years selection dyjaes gives
    0.70      0.52     0.48      0.47  0.45      0.44   0.43  0.43
    join  explore     city     just bookchicago dyjas
    0.41     0.40     0.39     0.39     0.36  0.33
```

Com isso é possível ter a informação de convite às pessoas para se lembrarem dos inícios dos anos e explorar a cidade de Chicago com o livro de Thomas Dyja, que está participando do programa *One Book One Chicago*. Essa informação foi vista graças às palavras descritas após o comando que utilizou correlação mínima de 0.3, sendo elas "dyja", "*beginning*" (começo), "r*emember*" (lembre-se), "*chipublib*" (abreviação para Biblioteca Pública de Chicago), "*years*" (anos), "*gives*" (dar), "*join*" (participe), "*explore*" (explore), "*city*" (cidade), "*bookchicago*" (parte do nome do programa).

Levando em consideração as escolas, foi feita a associação da palavra "*students*", que quer dizer estudantes em português com as outras palavras dos *tweets* usando valor de associação mínimo de 0.1. Assim é possível ver que alguns estudantes vão começar ou que é um bom dia de começar os estudos de introdução urbanos da cidade de Chicago através de um livro e os estudantes estão ou alguém está muito entusiasmado por isso. As palavras em inglês que foram associadas a "*students*" são "*introduce*" (introduzir), "*studies*" (estudos), "*urban*" (urbano), "*good*" (bom), "*excited*" (entusiasmado), "*today*" (hoje) e "*bookchicago*" (livro de Chicago). Os comandos e o resultado estão a seguir.

```
findAssocs(tdm, "students", 0.1)
$students
  introduce     studies       thats       urban        good     excited       today bookchicago
       1.00        1.00        1.00        1.00        0.53        0.34        0.32        0.11
```

O uso de associações entre as palavras dá informação prévia, mas é necessário que quem esteja observando os resultados tenha certo conhecimento ou interação com o assunto que está sendo trabalhado. Nesse caso, esse conhecimento viria de saber do que se trata o programa que está sendo estudado, o que livro está abordando e em qual cidade está ocorrendo esse processo. É um recurso que deve ser levado em consideração, pois traz informações relevantes, sendo simples e rápido de se usar.

4.4.5 *Word Cloud* (Nuvem de Palavras)

A nuvem de palavras é feita com as palavras de maior frequência nos *tweets*. Quanto maior a frequência da palavra, maior o tamanho da fonte da palavra que é apresentada.

Antes do processo de criação da nuvem de palavras, algumas palavras usadas como links de assunto (*hashtags*) foram retiradas, por não darem informação relevante e somente relacionarem os *tweets* ao programa *One Book One Chicago*. O passo inicial foi criar o conjunto de palavras a serem retiradas, denominado *OBOCstopwords*, logo após é criado um novo *corpus* com esses valores retirados, e por fim a matriz de palavras (‘*tdm’*) tende a ser rodada novamente, para assim a nuvem de palavras poder ser construída posteriormente. Os comandos utilizados para a formação da matriz de palavras estão abaixo.

```
OBOCstopwords = c("oboc","bookchicago","onebookonechicago")
mycorpus = tm_map(mycorpus, removeWords, OBOCstopwords)
tdm = TermDocumentMatrix(mycorpus, control = list(wordLengths = c(3, Inf)))
```

Para ser feita a nuvem de palavras é necessária instalação do pacote denominado ‘*wordcloud’*. Outro pacote que também foi usado foi o ‘*RColorBrewer’,* para a colocação de cores diversas na nuvem de palavras, fazendo com que visualmente fique mais fácil localizar os termos. Seguem os comandos utilizados.

```
library(wordcloud)
library(RColorBrewer)
m = as.matrix(tdm)
word.freq = sort(rowSums(m), decreasing = T)
wordcloud(words = names(word.freq), freq = word.freq, min.freq = 10, random.order = F, colors=brewer.pal(8, "Dark2"))
```

A frequência mínima utilizada nesse caso foi de 10, podendo ser vista no parâmetro da função ‘*wordcloud’* como ‘*min.freq’*. Dessa maneira, foi possível ver com clareza as palavras sem o gráfico ficar muito poluído. O resultado dessa nuvem de palavras aparece na Figura 11.

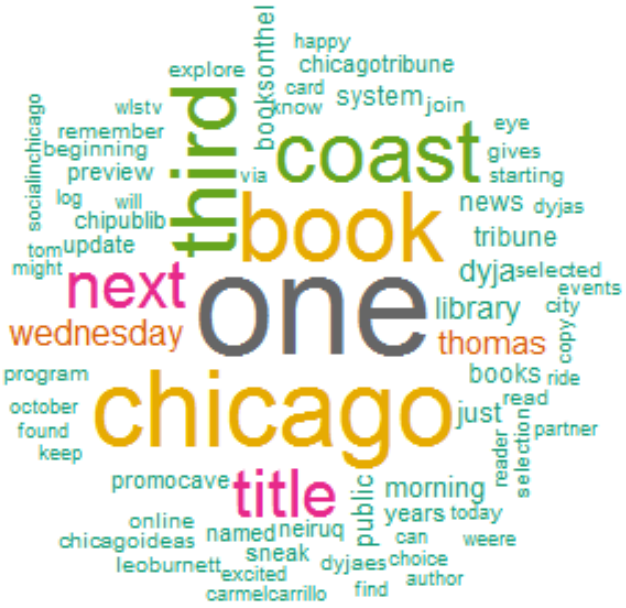

Figura 11: Nuvem de palavras com *tweets* coletados pelo programa *OBOC.*

É possível identificar na nuvem de palavras as palavras mais frequentes pelo seu maior tamanho e cores variadas que falam sobre o nome do programa e o nome do livro do autor *Thomas Dyja*: "*one*", "*book*", "*chicago*", "*third*", "*coast*", "*thomas*", "*dyja*", "*title*". Outras palavras em volta com menos frequência que são menores e com predominância de cor verde falam das notícias, da biblioteca de pública de Chicago, sobre leitura, sentimento, informações *online* entre outras sendo algumas aqui citadas: "*read*" (ler), "*excited*" (animado), "*library*" (biblioteca) e "*online*".

Atualmente é possível fazer nuvens de palavras online em alguns sites, digitando o seu texto ou mandando o endereço da página que deseja. Alguns desses sites podem ser encontrados no endereço da internet (EDGARCOSTA, 2016). Muitos desses sites ainda possibilitam que se mude o formato da nuvem de palavras e não fique somente com o círculo, sendo possível fazer uma nuvem de palavras como uma imagem que pode ser utilizada em alguma campanha publicitária, por exemplo.

4.4.6 Análise de *Cluster*

Nessa subseção foram usadas três técnicas de *clustering*: por matriz de distâncias (hierárquico), *k-means* e *around medoids*.

Para serem feitos os agrupamentos de palavras, também chamados de *Cluster* dos *tweets* encontrados pelo programa OBOC, deve-se usar a matriz de termos '*tdm'* somente com seus termos mais frequentes. Para os termos menos frequentes serem removidos é usada uma função chamada '*sparse'*. Sendo assim, a nova matriz '*tdm2new'* ficou com 33 termos. Foi fixada uma semente antes de começar essa análise para ser rodada todas as vezes e ter os mesmos resultados de grupos formados, com a função '*set.seed'*.

Após isso, uma matriz de distâncias entre as palavras é feita, sendo chamada de '*distMatrixnew',* por meio da função '*dist'*. Utilizando o método '*ward.D'*, foram feitos os *clusters*. A Figura 12 mostra o dendograma com as 33 palavras e a divisão que pareceu a mais correta visualmente (muitas combinações foram feitas com divisão de *clusters* para ver quais palavras ficariam juntas e isoladas), com 6 clusters. Utilizou-se a função '*plot'* para construir o dendograma e a função '*rect.hclust'* para adicionar as linhas de divisão dos *clusters* no gráfico, que faz uma linha de cima para baixo e divide com o número desejado. Isso é utilizado para se ter uma ideia visual de quantos *clusters* podem ser obtidos.

```
set.seed(122)
tdm2new = removeSparseTerms(tdm, sparse = 0.95)
mnew = as.matrix(tdm2new)
distMatrixnew = dist(scale(mnew))
fitnew = hclust(distMatrixnew, method = "ward.D")
plot(fitnew)
result2 = rect.hclust(fitnew, k=6)
```

No dendograma é possível ver assuntos identificados como o próximo livro do programa *OBOC*, com as palavras "*next*" (próximo), "*title*" (título), "*book*" (livro), "*chicago*". Outra informação que é possível de ver é que as palavras "*third*" e "*coast*" estão associadas, pois é o nome do livro participante do programa *OBOC*. As palavras contidas nos diferentes retângulos vermelhos formam os diferentes assuntos possíveis de serem identificados no dendograma.

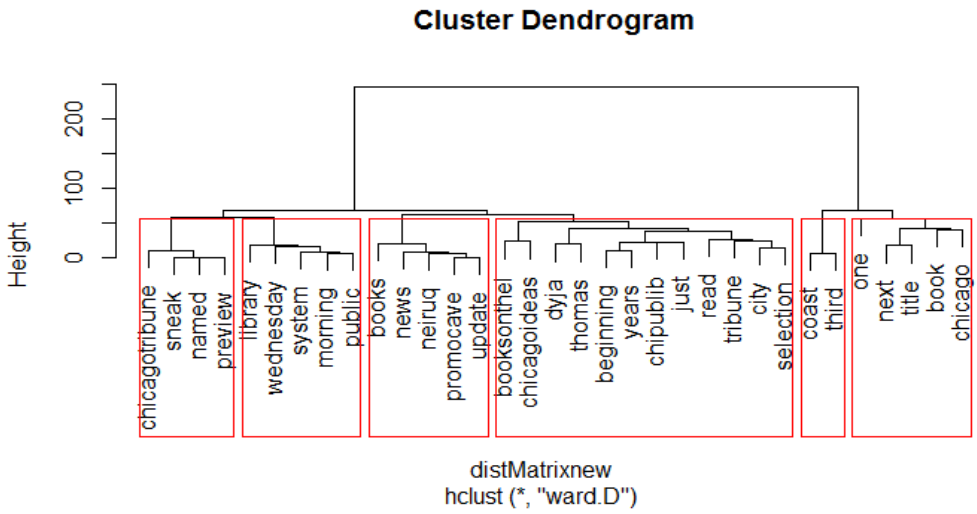

Figura 12: Dendograma dos *tweets* do programa *OBOC*

Seguindo para a segunda técnica de *clustering*, o *k-means*, os *tweets* foram agrupados novamente por assunto. Primeiramente a matriz com somente os termos mais frequentes antes calculada (‘*mnew’*) foi transposta e os *tweets* foram colocados em grupos de acordo com o número de *clusters* utilizado anteriormente (k=6). A função ‘*print’* mostrará alguns *tweets* de cada *cluster*.

```
m3 = t(mnew)
k = 6
kmeansResult = kmeans(m3,k)
for(i in 1:k){
  cat(paste("cluster", i, ": ", sep = ""))
  s = sort(kmeansResult$centers[i,], decreasing = T)
  cat(names(s)[1:6], "\n")
  print(head(banktextconv2[which(kmeansResult$cluster==i)]))
}
```

Na sequência são apresentados os seis *clusters* resultantes, mostrando primeiramente as seis palavras que os identificam e após um demonstrativo de *tweets* de cada *cluster*. Algumas palavras ou pontuações são irrelevantes para compreensão das frases, pois para ser usada a função ‘*print’* para mostrar os *tweets* não é possível utilizarmos o *corpus*.

Cluster 1 : beginning book coast dyja just third (em português: começo, livro, costa, dyja, somente, terceiro)

[1] "This year's @1book1chicago is \"The Third Coast\" by Thomas Dyja! Remember, the book is just the beginning. http://t.co/unJ3rV2YHQ"
[2] "This year's @1book1chicago is \"The Third Coast\" by Thomas Dyja! Remember, the book is just the beginning. http://t.co/unJ3rV2YHQ"

[3] "This year's @1book1chicago is \"The Third Coast\" by Thomas Dyja! Remember, the book is just the beginning. http://t.co/unJ3rV2YHQ"
[4] "RT @chipublib: This year's @1book1chicago is \"The Third Coast\" by Thomas Dyja! Remember, the book is just the beginning. http://t.co/unJ3rV<e2><80><a6>"
[5] "RT @chipublib: This year's @1book1chicago is \"The Third Coast\" by Thomas Dyja! Remember, the book is just the beginning. http://t.co/unJ3rV<e2><80><a6>"
[6] "RT @chipublib: This year's @1book1chicago is \"The Third Coast\" by Thomas Dyja! Remember, the book is just the beginning. http://t.co/unJ3rV<e2><80><a6>"

Nesse primeiro *cluster* é possível ver que o livro de Thomas Dyja, *Third Coast*, é somente o começo do programa *One Book One Chicago*, e que as pessoas que estão lendo o *tweet* podem esperar ainda mais livros e mais conteúdo, ou seja, um indício que o programa deve continuar com novos títulos de livros. Como exemplo, segue a tradução do *tweet* 1: "O livro deste ano para o programa *One Book One Chicago* é *Third Coast* do autor Thomas Dyja! Lembrando, este livro é somente o começo".

Cluster 2: third coast thomas chicago city just (em português: terceiro, costa, thomas, Chicago, cidade, somente)

[1] "@librarygrrrl @loather @booksNyarn I'm third coast but absurdly awake"
[2] "Want to read a book with the whole city of #Chicago? Pick up a copy of \"Third Coast\" by Thomas Dyja: http://t.co/0Ht2p2lEbM"
[3] "Want to read a book with the whole city of #Chicago? Pick up a copy of \"Third Coast\" by Thomas Dyja: http://t.co/0Ht2p2lEbM"
[4] "Want to read a book with the whole city of #Chicago? Pick up a copy of \"Third Coast\" by Thomas Dyja: http://t.co/0Ht2p2lEbM"
[5] "RT @1book1chicago: Thomas Dyja<e2><80><99>s The Third Coast is the 2015-2016 #OBOC selection! Join us as we explore Chicago: The City That Gives! http<e2><80><a6>"
[6] "RT @1book1chicago: Thomas Dyja<e2><80><99>s The Third Coast is the 2015-2016 #OBOC selection! Join us as we explore Chicago: The City That Gives! http<e2><80><a6>"

O segundo *cluster* é mais para falar sobre o livro do momento da cidade, o *Third Coast*, e que todos deveriam ler juntos, ou seja, chamando as pessoas para se juntarem em um grupo e explorarem a cidade de Chicago através da leitura. Um exemplo é o *tweet* 2: "Quer ler um livro com toda a cidade de Chicago? Pegue uma cópia do *Third Coast* de Thomas Dyja".

Cluster 3: booksonthel chicagoideas book dyja books thomas (em português: livros na linha de trem amarela, ideias de Chicago, livro, dyja, livros, thomas)

[1] "British Film Institute Film and Television Handbook 1998 by Eddie Dyja pb book http://t.co/aJ58kuTJ32 http://t.co/6ye5hDGMjr"
[2] "RT @1book1chicago: We know, we know, you *really* wanna know what the new #OBOC is! It's coming soon! While you wait, some past picks http:<e2><80><a6>"
[3] "A day long tour with presentations inspired by #OBOC book #StationEleven!... http://t.co/wmbRGcvP7m"
[4] "RT @1book1chicago: We know, we know, you *really* wanna know what the new #OBOC is! It's coming soon! While you wait, some past picks http:<e2><80><a6>"
[5] "RT @1book1chicago: #OBOC #ReadingSprints are back by popular demand this season on Twitter! Stay tuned for the schedule!"

[6] "RT @1book1chicago: We have an exciting lineup of #OBOC events both on our mainstage at Harold Washington & in *every single branch* of @chi<e2><80><a6>"

Olhando para o *cluster 3* se tem a ideia de que as pessoas estão ansiosas para saber qual será o novo livro do programa *One Book One Chicago* e que eventos desse programa estão por vir. Como exemplo, segue a tradução do *tweet* 2: "Nós sabemos, nós sabemos, você realmente quer saber qual o novo livro do *OBOC*! Vai chegar logo! Enquanto espera, procure algumas dicas no site sobre o programa".

Cluster 4: one chicago book title next coast (em português: um, Chicago, livro, título, próximo, costa)

[1] "#promocave Book News Update: 'Third Coast' to be next One Book, One Chicago title http://t.co/0TTs2IuMzy #books"
[2] "#promocave Book News Update: 'Third Coast' to be next One Book, One Chicago title http://t.co/0TTs2IuMzy #books"
[3] "#promocave Book News Update: 'Third Coast' to be next One Book, One Chicago title http://t.co/0TTs2IuMzy #books"
[4] "#promocave Book News Update: 'Third Coast' to be next One Book, One Chicago title http://t.co/0TTs2IuMzy #books"
[5] "#promocave Book News Update: 'Third Coast' to be next One Book, One Chicago title http://t.co/0TTs2IuMzy #books"
[6] "RT @neiruq: #promocave Book News Update: 'Third Coast' to be next One Book, One Chicago title http://t.co/0TTs2IuMzy #books"

O *cluster* 4 passa a atualização de livro do programa *OBOC*, dizendo que o próximo título de livro do programa é o *Third Coast* de Thomas Dyja. Um exemplo é o *tweet* 1: "Atualização de notícias de livro: '*Third Coast*' será o novo título do programa *One Book One Chicago"*.

Cluster 5:one book chicago third coast dyja (em português: um, livro, Chicago, Terceira, costa, dyja)

[1] "<e2><80><98>The Third Coast,<e2><80><99> by Dyja is the next One Book, One Chicago #OBOC This review always tainted my thoughts on book: http://t.co/ITmJrN7ZEv"
[2] "<e2><80><98>The Third Coast,<e2><80><99> by Dyja is the next One Book, One Chicago #OBOC This review always tainted my thoughts on book: http://t.co/ITmJrN7ZEv"
[3] "<e2><80><98>The Third Coast,<e2><80><99> by Dyja is the next One Book, One Chicago #OBOC This review always tainted my thoughts on book: http://t.co/ITmJrN7ZEv"
[4] "<e2><80><98>The Third Coast,<e2><80><99> by Dyja is the next One Book, One Chicago #OBOC This review always tainted my thoughts on book: http://t.co/ITmJrN7ZEv"
[5] "So happy that The Third Coast was selected for @chipublib's One Book One Chicago. Here are my 2 photos from the book http://t.co/tZ1zciQFve"
[6] "So happy that The Third Coast was selected for @chipublib's One Book One Chicago. Here are my 2 photos from the book http://t.co/tZ1zciQFve"

As informações encontradas no *cluster 5* são sobre *reviews* que estragaram a leitura de outras pessoas, pois podem ter falado demais ou falado mal sobre o livro em questão (*Third Coast*). Além disso, também é visto que algumas pessoas estão felizes que o livro *Third Coast*

foi selecionado para o programa e também tiraram fotos para mostrar o livro, isso também serve como divulgação do livro. Dois exemplos foram retirados desse *cluster* que são – *tweet 1*: "*The Third Coast* escrito por Thoma Dyja é o novo livro do programa *One Book One Chicago*. Este *review* sempre estraga meus pensamentos sobre o livro" e *tweet 5:* "Tão feliz que *Third Coast* foi selecionado para *One Book One Chicago* . Aqui estão minhas fotos do livro".

Cluster 6: just book booksonthel city chipublib beginning (em português: somente, livro, livros na linha de trem amarela, cidade, biblioteca pública de Chicago, começo)

[1] "The book is just the beginning....and we're ready to share it today! #OBOC"
[2] "RT @chipublib: The book is just the beginning....and we're ready to share it today! #OBOC"
[3] "RT @chipublib: The book is just the beginning....and we're ready to share it today! #OBOC"
[4] "What's this Burnetter doing? Oh, just leaving copies of @1book1chicago on the seats #booksontheL #OBOC #leoburnett http://t.co/ryLbZtwl1Z"
[5] "What's this Burnetter doing? Oh, just leaving copies of @1book1chicago on the seats #booksontheL #OBOC #leoburnett http://t.co/ryLbZtwl1Z"
[6] "Wow! A book just for me to borrow! Found at Paulina Brown Line. :) I love my city. #BooksontheL #OBOC http://t.co/xdavESzHcZ "

Nesse último *cluster 6* é mostrado que alguns usuários do *Twitter* estão prontos para saber qual será o novo título do programa *One Book One Chicago* e que algumas pessoas estão deixando cópias desse livro pelas linhas de trem amarela (*booksonthel* é uma maneira de falar livros na linha de trem amarela). Dois exemplos são dados com o *twitter 1*: "O livro é somente o começo...e nós estamos prontos para dividi-lo hoje!" e *twitter 5: "*O que *burnetter* está fazendo? Oh, somente deixando cópias do livro do programa *One Book One Chicago* nos assentos dos trens*".* Nesse segundo exemplo aparece a palavra "*burnetter"* que diz respeito a Leo Burnett, um executivo muito influente do ramo de propaganda.

É possível ver que alguns *clusters* compartilham informação entre si como o *Cluster 4* e *Cluster 5* que falam que o livro *Third Coast* foi selecionado como sendo o próximo livro do programa *One Book One Chicago*.

Nuvens de palavras foram feitas para mostrar os seis diferentes *clusters* de forma visual. Os comandos são iguais aos utilizados na nuvem de palavras feita anteriormente, mas agora terá um *loop* para que se possa pegar o conjunto de palavras de cada *cluster* por vez. O comando '*par'* foi anteriormente rodado para ter a janela dos gráficos dividida em seis partes começando da esquerda para à direita e de baixo para cima. Os comandos seguem abaixo e a Figura 13 mostra as seis nuvens de palavras resultantes.

```
par(mfrow=c(2,3))
for(j in 1:6){
  a1 = mycorpus[which(kmeansResult$cluster==j)]
  tdm1 = TermDocumentMatrix(a1, control = list(wordLengths = c(3, Inf)))
  m1 = as.matrix(tdm1)
  word.freqa = sort(rowSums(m1), decreasing = T)
  wordcloud(words = names(word.freqa), freq = word.freqa, min.freq = 5, random.order = F, colors=brewer.pal(8, "Dark2"))
}
```

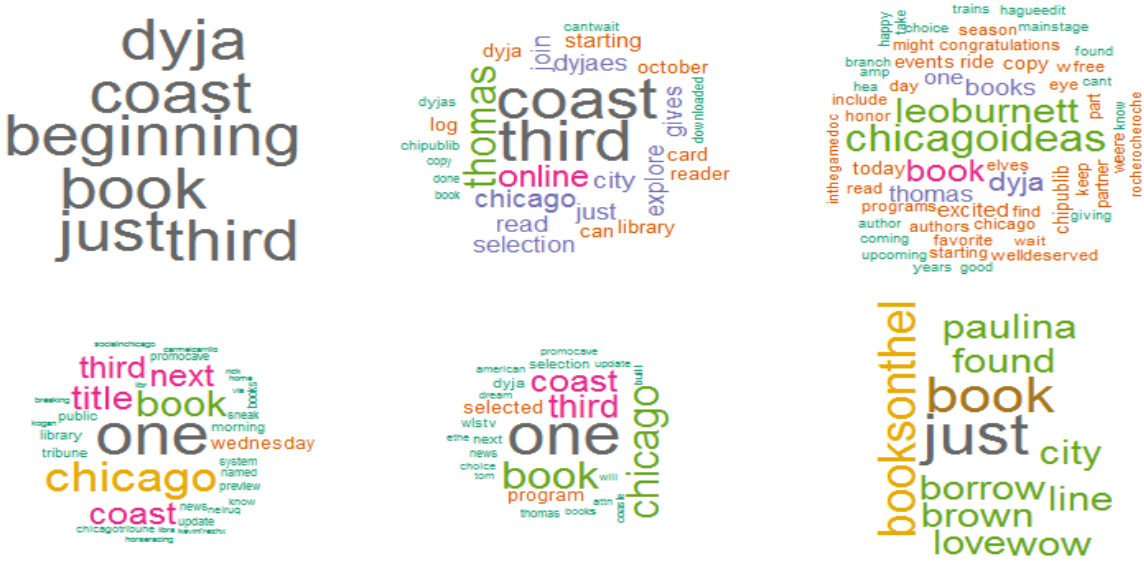

Figura 13: *Wordclouds* de cada *k-means cluster*.

As nuvens de palavras apresentam visualmente o resultado mostrado anteriormente das palavras divididas em seis *clusters*, mas de forma gráfica. É necessário dizer também que algumas palavras com baixa frequência são omitidas das nuvens por não ter espaço nos gráficos.

Com todos os *clusters* separados, é possível que os gestores do programa *One Book One Chicago* possam ver o que as pessoas estão falando de determinado assunto e assim buscar melhorias, seja na divulgação do programa, incentivo para que as pessoas deem mais opiniões sobre o programa nas redes sociais, entre outras, sempre na busca de um programa melhor, que ajude as pessoas da cidade a lerem mais e terem mais conhecimento.

Para a última análise de *cluster* foi feito um agrupamento *Around Medoids* (em torno de centróides) com os *tweets* do programa *OBOC*. Essa análise de *cluster* é uma versão mais robusta do *k-means* calculado anteriormente. Nesse caso foi usada a distância *Manhattan* ao invés da *Euclidiana*. A diferença é que pode medir a distância não do modo reto, mas como se seguisse

um caminho alternativo. Ambas usam a matriz de distância dos *tweets* (onde cada coluna representa um *tweet*), mas a fórmula para a distância *Manhattan* é mais simples que para a distância *Euclidiana*, pois usa somente a soma das diferenças entre os *tweets*.

Para ser calculado o número de *clusters* foi usado o pacote '*fpc'* e a função '*pamk'*. O resultado foi o número de dez *clusters*, sendo assim foram apresentadas as palavras relacionadas a cada *cluster*, podendo ter *clusters* sem palavras, pois a frequência de palavras de determinado *cluster* não foi o bastante para serem vistos como identificadores. Abaixo seguem os comandos utilizados e as palavras para cada *cluster*.

```
library(fpc)
pamResult = pamk(m3, metric ="manhattan")
k1 = pamResult$nc
pamResult = pamResult$pamobject
for(i in 1:k1) {
  cat("cluster", i, ": ",
      colnames(pamResult$medoids)[which(pamResult$medoids[i,]==1)],"\n")
}
```

```
cluster 1 :  coast third
cluster 2 :
cluster 3 :  books chicago coast neiruq news next promocave third title update
cluster 4 :  book chicago coast next third title
cluster 5 :  chicago city coast selection third thomas
cluster 6 :  beginning book chipublib coast dyja just third thomas years
cluster 7 :  book coast next third title tribune
cluster 8 :  book chicago title
cluster 9 :  book coast library morning next public system third title wednesday
cluster 10 :  book chicago chicagotribune coast named next preview sneak third title wednesday
```

Foram feitas nuvens de palavras para se ter a ideia de cada *cluster* visualmente, sendo nesse caso feito 10 nuvens de palavras, uma relacionada com cada *cluster*. Os passos são os mesmos feitos anteriormente para a primeira nuvem de palavras, mas agora se precisa usar a função '*par'* para dividir a janela de gráfico e ser possível dispor os 10 diferentes gráficos, além de termos um *loop* utilizando o comando *'for'*. Os comandos são apresentados na sequência, e após a Figura 14 apresenta todas as *wordclouds*, tendo a ordem de cada *cluster* seguida de maneira que os primeiros cinco *clusters* representados da esquerda para à direita na primeira linha e os outros cinco *clusters* representados da esquerda para à direita na segunda linha. Com esse processo também é possível ver as palavras que podem não ter aparecido para alguns *clusters* como no caso do *cluster* 2 pela falta de frequência das palavras.

```
par(mfrow=c(2,5))
for(j in 1:10){
  cl <- which( pamResult$clustering == j )
  tdmk <- t(m3[cl,])
  v = sort(rowSums(tdmk), decreasing=TRUE)
  d = data.frame(word=names(v), freq=v)
  wordcloud(d$word, d$freq, min.freq=5,
          random.color=TRUE,colors="black")
}
```

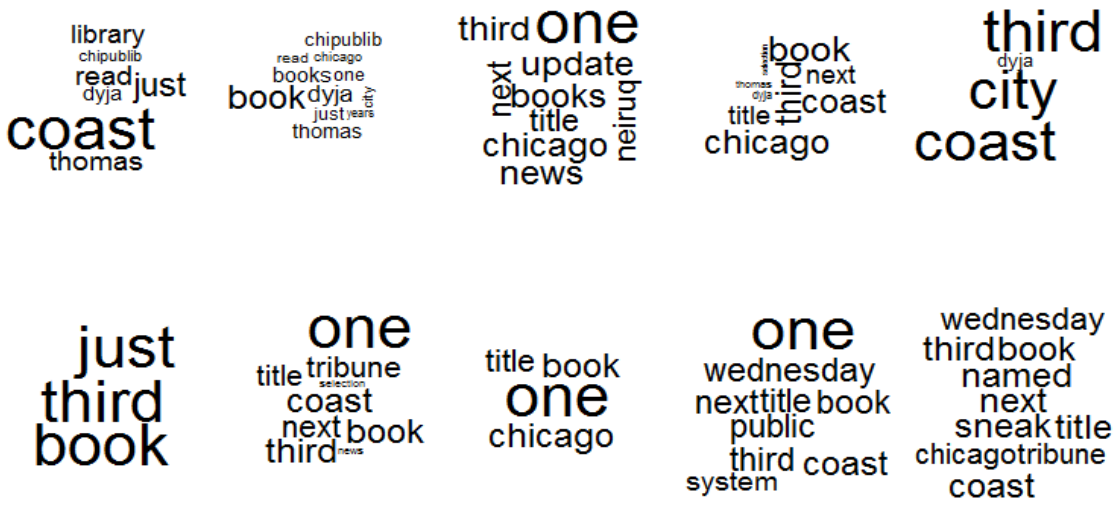

Figura 14: *Wordclouds* de cada *cluster around medoids*

As nuvens de palavras mostram graficamente o que foi resultado das palavras de cada *cluster*. Palavras com frequência menor que não aparecem, pois não cabem em cada nuvem de palavras devido ao tamanho do gráfico combinado que é gerado.

Após isso, foi construído um gráfico para ser observado como os *clusters* ficam divididos e tirar algumas conclusões. O pacote usado para a construção do gráfico foi '*cluster'*. Os comandos seguem abaixo e logo após é apresentado o gráfico na Figura 15.

```
library(cluster)
cusplot(pamResult, col.p = pamResult$clustering)
```

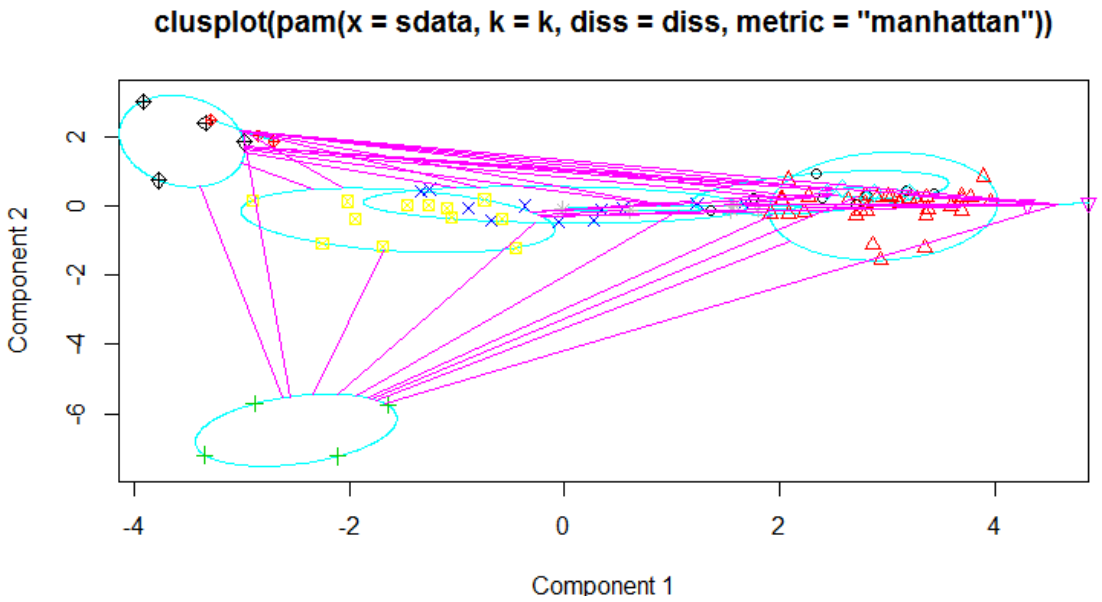

Figura 15: Gráfico de *Clusters around medoids* dos *tweets* do programa *OBOC*.

A Figura 15 mostra os dez diferentes *clusters* identificados com diversos tipos de ícones, como, por exemplo, triângulos. Os círculos maiores em azul mostram as partições (*around medoids*). Percebe-se que há alguma sobreposição entre *clusters*, o que quer dizer que os *clusters* tem informações semelhantes, com mesmos *tweets* entre si, como já comentado anteriormente. Também é possível ver que mais ou menos quatro grupos se diferem mais entre si, observando os círculos da esquerda abaixo, da esquerda acima, da esquerda ao meio e da direita acima no gráfico, com muitas sobreposições entre si. Os dois componentes mais explicativos usados para a construção do gráfico explicam 38.44% da variabilidade entre *clusters*.

Como visto anteriormente na análise de *cluster* por *k-means* há muita informação que é dividida entre os diferentes *clusters* o que pode, às vezes, fazer com que muita informação fique conjunta e seja preciso olhar em mais de um *cluster* se quiser, por exemplo, saber se as pessoas estão felizes com a escolha do livro *Third Coast* para a temporada de 2015 do programa *One Book One Chicago.*

4.4.7 *Topic Modelling*

Topic Modelling ou Modelagem por Tópicos é uma técnica utilizada somente para ter uma ideia inicial do que os textos estão tratando. Não funciona tão bem como técnicas de *cluster*,

pois muitas vezes escolhe palavras que não dizem muito sobre os textos. Mesmo utilizando uma semente no *Software R*, as palavras mudam de acordo com o número de vezes que é rodado o programa.

O pacote utilizado foi o '*topicmodels'*. Primeiro se acham os tópicos com a função '*LDA'* onde foi utilizado o número de seis tópicos, como o número de *clusters* encontrados no *k-means*, e depois se definem quantas palavras serão mostradas em cada tópico, sendo utilizadas seis palavras. Os comandos e o resultado estão abaixo.

```
library(topicmodels)
lda = LDA(dtm, k=6)
term = terms(lda,6)
term
```

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
[1,]	"thomas"	"booksonthel"	"one"	"one"	"chicago"	"online"
[2,]	"third"	"chicagoideas"	"chicago"	"book"	"one"	"read"
[3,]	"coast"	"city"	"book"	"chicago"	"book"	"coast"
[4,]	"dyja"	"leoburnett"	"title"	"coast"	"next"	"third"
[5,]	"chipublib"	"chicago"	"coast"	"third"	"third"	"library"
[6,]	"just"	"book"	"third"	"news"	"coast"	"reader"

No primeiro tópico tem-se que *Third Coast*, o livro de *Thomas Dyja*, estará presente nas bibliotecas públicas de Chicago. Palavras em inglês: "*thomas*", "*third*"(terceira), "*coast*"(costa), "*dyja*", "*chipublib*"(abreviação de biblioteca pública de Chicago), "*just*"(somente).

Nesse segundo tópico é possível ver que a cidade de Chicago tem algumas ideias sobre os livros deixados pelos trens para motivação da leitura. Palavras em inglês: "*booksonthel*" (livros na linha de trem amarela), "*chicagoideas*" (ideias de Chicago), "*city*" (cidade), "*leoburnett*" (*hashtag* para Leo Burnett), "*chicago*", "*book*"(livro).

O terceiro tópico conta que o livro do programa *OBOC* que está vigente no momento é intitulado *Third Coast*. Palavras em inglês: "*one*"(um), "*chicago*", "*book*" (livro), "*title*" (título), "*coast*" (costa), "*third*" (terceira).

No quarto tópico é visto que saiu notícias sobre o próximo livro do programa *OBOC* que é o *Third Coast*. Palavras em inglês: "*one*" (um), "*book*" (livro), "*chicago*", "*coast*" (costa), "*third*" (terceira), "*news*" (novidades).

O quinto tópico dá a informação que o próximo livro do programa *OBOC* será o *Third Coast*. Palavras em inglês: "*chicago*", "*one*" (um), "*book*" (livro), "*next*" (próximo), "*third*" (terceira), "*coast*" (costa).

Olhando para o sexto tópico temos que é possível ler o livro *Third Coast* através de um leitor *online* da biblioteca. Palavras em inglês: "*online*", "*read*" (ler), "*coast*" (costa), "*third*" (terceira), "*library*" (biblioteca), "*reader*" (leitor).

É possível perceber, como dito anteriormente, que a técnica de *Topic Modelling* é utilizada somente para se ter uma ideia rápida, e não muito precisa, sobre os diferentes tópicos que os textos contêm. Levando em consideração também a necessidade do conhecimento prévio sobre o assunto para poder tirar essas conclusões do que cada tópico aborda.

4.5 Pós-Processamento

A limpeza, integração e seleção dos dados são muito importantes para o começo da análise, assim como a armazenagem e preparação dos dados, pois é a partir desses passos que será possível fazer a análise do *Text Mining* corretamente. Dados que não possuíam nenhuma informação do programa ou do livro relacionado, um dos problemas enfrentados pelos gestores do programa *OBOC*, foram excluídos nessa primeira parte dos resultados.

Conhecendo as palavras mais frequentes que apareceram nos *tweets* foi possível tirar conclusões de quais assuntos foram citados nos *tweets*, verificando que a análise está indo de acordo com o esperado, que seria saber que os *tweets* estão mesmo falando do assunto pesquisado; o programa *OBOC*.

Com o *Text Mining*, foi possível determinar, com as associações de palavras, qual seria o próximo livro do programa *OBOC*, quando seria lançado, do que se tratava, além de que haviam pessoas entusiasmadas para lerem o livro e utilizá-lo como instrumento de estudo.

A nuvem de palavras foi uma maneira mais visual de obter informações sobre o que os *tweets* estavam falando, organizando o conhecimento prévio visto na frequência de palavras e associações entre palavras.

Três diferentes métodos de *cluster* foram empregados, além de uma modelagem por tópicos, para saber sobre o que os *tweets* estavam falando sobre o programa. Essas análises

dividiram os *tweets* em diferentes grupos, cada um com determinado tópico. Isso foi realizado afim de que os gestores do programa pudessem ver o que as pessoas estavam falando sobre determinado assunto, como por exemplo, divulgação do livro *Third Coast*.

A primeira análise de *cluster* feita com matriz de distâncias utilizou o dendograma para dividir os *tweets* em seis grupos de palavras, onde os tópicos achados não foram tão interessantes e significativos, como por exemplo, um deles era somente o nome do livro *Third Coast*. Na segunda análise de *cluster*, utilizando o método por *k-means*, os resultados foram melhores em relação aos diferentes tópicos, pois se obteve tópicos bem distribuídos, como por exemplo, o *cluster* 5 que tinha como assunto as opiniões das pessoas sobre o livro *Third Coast* e o sentimento de saber que esse livro tinha sido escolhido para participar do programa *OBOC*. O último método de *cluster, around medoids*, utilizou a matriz de distâncias *Manhattan* ao invés da *Euclidiana*, utilizada na primeira análise. Como resultado, obtiveram-se dez tópicos diferentes para os *tweets*, mas como a primeira análise, não gerou muita informação diferente sobre os *tweets*. A modelagem por tópicos não dividiu tão bem os grupos dos *tweets* como as análises de *cluster*, mas funciona como uma maneira rápida de divisão de tópicos, sendo nesse caso achados seis diferentes tópicos.

Com a utilização da técnica de *Text Mining* os gestores do programa *OBOC* puderam buscar nas redes sociais as opiniões das pessoas e assim utilizarem essas informações para estratégias de melhoria do programa, novidades, atuação, divulgação, entre outros, a fim de darem cada vez mais uma melhor experiência de leitura para a população residente da cidade de Chicago e influenciar a leitura.

5. CONSIDERAÇÕES FINAIS

A utilização da técnica de *Text Mining* é de grande importância para obter informações sobre textos sem uma leitura prévia. Sendo assim, é possível obter informação dos textos e utilizá-los como auxílio na procura de melhorias para algum tipo de produto, empresa, conhecimento sobre algum assunto que possa ser interessante para quem utiliza a técnica.

Este trabalho apresentou o *Text Mining*, expondo todas suas etapas e seu uso aliado com o *Software* R. Alguns exemplos de aplicação foram mostrados, como na área alimentícia e de esporte, mostrando a diversidade de áreas que a técnica de *Text Mining* abrange.

O *Software* R, que foi utilizado neste trabalho, é hoje um dos principais *softwares* para análise de textos, pois contém muitos pacotes voltados para essa área, além de ser de uso gratuito. Nos últimos anos, ganhou novos pacotes para auxílio no *Text Mining*, isso, juntamente com os pacotes que já possuía fez com que a análise feita nesse *software* seja muito informativa.

Os dados utilizados para aplicação do *Text Mining* foram obtidos da rede social *Twitter*, buscando dados sobre o programa *OBOC*. O intuito dessa análise era primeiramente aprender sobre a aplicação do *Text Mining* utilizando o *Software* R, tendo como objetivo secundário obter informações relevantes para o programa *One Book One Chicago*.

Entre as diferentes análises utilizadas, a que se mostrou mais significativa para o problema de achar os diferentes tópicos para as pessoas do programa *One Book One Chicago* foi a análise de *cluster k-means*. Essa análise apresentou mais informação relevante, além de ser um processo mais rápido e fácil de entender em relação às outras análises realizadas.

As técnicas de *Text Mining* são bem viáveis nos dias de hoje, mas não totalmente automatizadas. Pessoas com algum conhecimento sobre o assunto que está sendo estudado são necessárias para a análise, pois muitos processos contam ainda com decisões importantes para obter resultados relevantes. Neste trabalho, era necessário que quem estivesse fazendo a análise de *Text Mining* soubesse sobre o programa de leitura de Chicago e o livro participante.

A análise textual cresceu bastante nos últimos anos, pois cada vez mais as pessoas usam a internet para transmitir informações em texto, e isso, muitas vezes, pode ser uma informação muito valiosa para os gestores. Atualmente, um grande nicho de informação pode ser encontrado nas diversas redes sociais, sendo gratuita e de acesso público. Desta forma, obter esses dados e utilizar o *Text Mining* resultará em informação consistente para propor novas ideias, tomar decisões e ajustar os negócios de acordo com o público alvo.

6. REFERÊNCIAS BIBLIOGRÁFICAS

AGGARWAL, C. C.; ZHAI, C. X. *A Survey of Text Clustering Algorithms*. 2012.

DIXON, M. *An Overview of Document Mining Technology*. 1997.

EDGARCOSTA. Nuvem de Palavras, ferramentas *online* – Tecnologia/Educação/*Lifestyle* - Disponível em: <http://www.edgarcosta.net/informatica/web-informatica/ferramentas-on-line-para-criar-nuvens-de-palavras/>. Acesso em 19/05/2016.

FEINERER, I.; HORNIK, K; MEYER, D. *Text Mining Infrastructure in R.* Journal of Statistical Software, 2008.

FELDMAN, R.; DAGAN, I. *Knowledge Discovery in Textual Databases (KDT)*. Barllan University, Rmat-Gan, Israel, 1995.

GODFREY, D.; JOHNS, C.; SADEK, C.; MEYER, C.; RACE, S. *A Case Study in Text Mining: Interpreting Twitter Data From World Cup Tweets*. 2014.

GRUPTA, V.; LEHAL, G. S. *A Survey of Text Mining Techniques and Applications*. Journal of Emerging Technologies in web intelligence, 2009.

HE, W.; ZHA, S.; LI, L. *Social Media competitive analysis and text mining: A case study in the pizza industry*. International Journal of Information Management, 2012.

HOESCHL, H. C; BUENO, T. C. D.; BORTOLON, A.; MATTOS, E.; RIBEIRO, M. S. *AlphaThemis - Do texto ao conhecimento*. Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina. Campus Universitário, Trindade, Florianópolis, SC, Brasil, 2002.

MATHIAK, B.; ECKSTEIN, S. *Five Steps to Text Mining in Biomedical Literature*. Technische Universität Braunschweig, 2004.

OLHAR DIGITAL – o que acontece na *internet* a cada minuto. Disponível em: <http://olhardigital.uol.com.br/noticia/veja-o-que-acontece-na-internet-a-cada-minuto/36126>. Acesso em 20/05/2016.

SILVA, G. L. A. *Text Mining, um estudo a partir da rede social Twitter*. Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.

TAN, A. H. *Text Mining: the state of the art and the challenges*. Kent Ridge Digital Labs, 1999.

WIVES, L. K. *Tecnologias de descoberta de conhecimento em textos aplicadas à Inteligência Competitiva.* Exame de qualificação – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002.

ZH NOTÍCIAS – Assalto a postos de combustível e furto de carros em Santa Maria. Disponível em: <http://zh.clicrbs.com.br/rs/noticias/noticia/2016/06/dupla-suspeita-de-assalto-a-postos-de-combustiveis-e-furtos-de-carros-e-presa-em-santa-maria-6022249> Acesso em: 16/06/2016.

ZHAO, Y. *Twitter Data Analysis with R*. Monash University, Melbourne, 2016.