Universidade Federal de Pernambuco
Centro de Informática

Graduação em Ciência da Computação

Business Intelligence e Análise de Sentimentos no Contexto de Redes Sociais Online

Leonardo José de Andrade Costa Santos

Trabalho de Graduação

Recife
Novembro de 2015

Universidade Federal de Pernambuco
Centro de Informática

Leonardo José de Andrade Costa Santos

Business Intelligence e Análise de Sentimentos no Contexto de Redes Sociais Online

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: *Prof. Dr. Fernando da Fonseca de Souza*

Recife
Novembro de 2015

Agradecimentos

Primeiramente, gostaria de agradecer à Universidade Federal de Pernambuco e, especialmente, a todos que compõem o Centro de Informática, centro de excelência e destaque internacional. Durante toda a minha graduação, conheci docentes, funcionários e alunos apaixonados pelo que fazem e que sempre me motivaram, inspiraram e contribuíram diretamente na busca dos meus objetivos e interesses.

Ao grupo PET-Informática, do qual tive o prazer de participar por 3 anos. Com o Programa de Educação Tutorial, tive a oportunidade de estar em contato com colegas brilhantes que tornaram-se amigos que levarei para o resto da vida. O programa me permitiu sair da condição de expectador e vivenciar de forma ampla a graduação em Ciência da Computação, contribuindo para melhorar o curso e com a formação de outros colegas. Através de atividades de pesquisa, ensino e extensão, o programa contribuiu diretamente com meu enriquecimento cultural e intelectual, me permitindo crescer em vários aspectos. Assim, foi um prazer ter sido membro deste grupo, que considero uma família.

Ao professor, orientador e tutor Dr. Fernando da Fonseca de Souza, que tem brilhantemente me orientado desde o início da minha graduação, quando ingressei no grupo PET. Um ano após conhecê-lo, tive a honra de ser seu aluno na disciplina de Gerenciamento de Dados e Informação. Com a disciplina, o professor Fernando me introduziu à área de bancos de dados e gerenciamento de informação, fazendo com que fosse despertado em mim um interesse enorme neste ramo tão importante da ciência da computação. Graças à base sólida que me foi proporcionada, consegui meu primeiro emprego na área, como engenheiro de software do *Information Platform Group* da Microsoft, onde poderei por em prática os conhecimentos que adquiri com as várias disciplinas que cursei com este brilhante mestre.

À minha turma, que esteve comigo durante todos estes anos e com quem aprendi muito. Conviver com pessoas tão inteligentes certamente contribuiu fortemente para que eu pudesse maximizar meu aprendizado durante todos os momentos da minha graduação. Em especial, agradeço a Larissa, Maria Gabriela e Marina, minhas grandes amigas e colegas de projetos, com quem compartilhei vários momentos de desespero e cansaço, mas também muitos outros de muita alegria.

Por último, a toda a minha família. Especialmente, à minha avó Zélia e meus pais, Katia e João, que sempre me amaram e apoiaram em todas as minhas decisões. Além disso, me proporcionaram a melhor educação que eu poderia ter tido, permitindo que eu iniciasse e concluísse esta graduação. Graças aos seus esforços e suporte, consegui chegar até aqui e conquistar tudo que consegui. Levarei todos os seus ensinamentos por toda minha vida.

Resumo

A utilização de redes sociais tem crescido de forma assustadora nos últimos anos, o que contribuiu para um aumento exponencial na produção de informações na Web. Este crescimento tem atraído o interesse de diversas organizações por consistir numa oportunidade de investigar o que seus consumidores falam sobre suas marcas e obter informações sobre estes clientes, visto que uma análise eficaz deste tipo de informação pode ajudar a guiar decisões corporativas. Este trabalho tem como objetivo a realização de um estudo de técnicas e ferramentas de extração de largas quantidades de dados não estruturados oriundos de redes sociais online. Será discutido como este tipo de informação pode ser recuperado e como ela pode ser útil em diferentes contextos. Aliado ao processo de extração de dados, o trabalho seguirá explorando algoritmos de análise e tratamento de dados. Em particular, técnicas de análise de sentimentos, que possuem o objetivo de avaliar sentimentos expressos em fragmentos de texto. Por fim, será mostrado um estudo de caso desenvolvido que ilustra como soluções de *Business Intelligence* e análise de sentimentos podem ser combinadas dentro do contexto de redes sociais para diversos tipos de análise que podem ser extremamente úteis para corporações.

Palavras-chave: extração de dados, big data, integração de dados, dados não estruturados, dados sociais, mineração de opinião.

Abstract

The usage of social networks has grown substantially in the past years and this has contributed to an exponential increase of the production of information on Web. This growth has attracted the interest of several organizations since it is an opportunity to investigate what their customers talk about their brands and to obtain meaningful information about these clients. This is particularly relevant because an efficient analysis of this kind of information can help enormously to guide corporate decisions. This work intends to make a study of the available techniques and tools to extract large amounts of non-structured data from online social networks. It will discuss how this sort of information can be retrieved and how it can be useful in several different contexts. Along with this data extraction process, the work then follows with exploring data analysis algorithms. In particular, sentiment analysis techniques, which are useful for classifying sentiments expressed in text fragments. Finally, the work will show a developed case study that illustrates how Business Intelligence and sentiment analysis can be combined in the context of online social networks to perform a number of different types of analysis that can be valuable to corporations.

Keywords: data extraction, big data, data integration, non-structured data, social data, opinion mining.

Sumário

CAPÍTULO 1

Introdução

Na última década, observou-se um crescimento imenso da utilização de redes sociais no Brasil e no mundo. Este crescimento tem sido particularmente acelerado com a inclusão digital, popularização de *smartphones* e acesso à rede de baixo custo. *Websites* como Twitter[1], Facebook[2], LinkedIn[3], Instagram[4], Pinterest[5], entre vários outros, têm atraído usuários, que se conectam com outros membros e se comunicam de diferentes formas sobre os mais variados temas, sentindo-se livres para expressar suas opiniões e potencialmente influenciar outros usuários (POLONI; TOMAÉL, 2014).

Essa ampla utilização de redes sociais traz consigo uma oportunidade para que empresas possam entender as necessidades de seus consumidores, descobrir o que tem sido falado sobre suas marcas, monitorar seus concorrentes, traçar perfis de seus clientes, entre outras estratégias relevantes para adquirir informações sobre o seu público (WEBER, 2009). Assim, encontrar, processar e armazenar dados relevantes no mercado de mídias sociais tem sido uma das maiores preocupações de organizações nos dias atuais, que têm investido cada vez mais em soluções com esta finalidade.

Para possibilitar diferentes análises da informação compartilhada nestas diferentes mídias sociais, é necessário realizar a coleta dos dados que são veiculados nestas plataformas. A fim de satisfazer essa necessidade de extrair este tipo de informação, atualmente, as principais redes sociais provêm interfaces de programação (*Application Programming Interfaces* – API), que permitem a captura total ou parcial de seus dados (FRANÇA et al., 2014). Este tipo de ferramenta de consumo viabiliza o uso de dados de redes sociais na alimentação de diversos tipos de aplicação.

Dispondo dos dados coletados, é preciso analisá-los para que se possa inferir informações relevantes. Devido à rápida produção de informação que se observa em redes sociais, uma análise manual deste tipo de dados torna-se inviável, fazendo-se necessária a busca por formas automáticas de classificar e interpretar textos publicados nestas mídias. Uma área de pesquisa neste contexto que vem ganhando atenção da academia é a de análise de sentimentos ou mineração de opinião, que se refere ao tratamento computacional de um texto com o intuito de identificar se este representa uma sentença positiva, negativa ou neutra sobre certo tópico (COSTA et al., 2012). Assim, conceitos de análise de sentimentos podem contribuir fortemente para uma compreensão mais detalhada de dados de redes sociais.

Para entender de forma completa os dados analisados, também é preciso incorporar dados temporais e ferramentas de visualização apropriadas. Soluções de *Business Intelligence*

[1] https://www.twitter.com/

[2] https://www.facebook.com/

[3] https://www.linkedin.com/

[4] https://www.instagram.com/

[5] https://www.pinterest.com/

(BI) (LUHN, 1958), que abrangem técnicas e ferramentas para transformação de dados puros em informação útil e relevante para análises de negócios, já conquistaram seu espaço entre grandes organizações por suprirem essa necessidade e continuam crescendo com um dos maiores índices no mercado de software (SALLAM et al., 2015). Este tipo de solução também pode ser usado para visualizar dados e explorar tendências em mídias sociais. Por meio da análise e geração de relatórios em tempo real, observações mais sofisticadas são possíveis, permitindo às organizações construir correlações que a observação humana por si só não seria capaz de replicar.

Dessa forma, é possível observar que a extração de dados de mídias sociais, combinada com a análise de sentimentos e soluções de Business Intelligence podem contribuir imensamente para o mundo corporativo para a análise de dados sociais e todas as suas vantagens associadas. Sendo assim, torna-se importante a realização de um estudo sobre redes sociais online, métodos e técnicas disponíveis de extrair dados deste tipo de plataforma e como as áreas previamente mencionadas podem ser combinadas em uma arquitetura que permita análises relevantes para organizações.

1.1 Objetivos

O objetivo deste trabalho é fazer um estudo sobre redes sociais online e técnicas computacionais que podem ser utilizadas dentro desse contexto para viabilizar e otimizar análises úteis a diversos tipos de organizações. Especificamente, pretende-se explorar o processo de extração de dados das principais plataformas sociais atuais e como técnicas de análise de sentimentos e soluções de Business Intelligence podem ser combinadas para permitir consulta e visualização de informações relevantes a partir destes dados.

1.2 Estrutura do Trabalho

Este trabalho está dividido em 6 capítulos, incluindo este capítulo introdutório. No Capítulo 2, será abordado com maior profundidade o conceito de redes sociais, detalhando cada uma das principais plataformas atuais, com foco em suas peculiaridades e como é possível extrair dados de cada uma delas. Em seguida, o Capítulo 3 discorrerá sobre análise de sentimentos e como a área pode ser útil para classificação de conteúdo social. O Capítulo 4 realizará um estudo sobre de conceitos de *Business Intelligence*, soluções disponíveis no mercado e como a área pode ser combinada com dados de redes sociais *online*. Após extenso embasamento teórico, o Capítulo 5 apresentará uma implementação do autor de uma arquitetura baseada em extração de dados sociais, análise de sentimentos e *Business Intelligence* para ilustrar sua eficácia, mostrando os resultados obtidos. Por fim, o Capítulo 6 expõe as conclusões obtidas, mostrando os resultados da pesquisa, implementação realizada e desafios encontrados na área, bem como as limitações e sugestões de trabalhos futuros relacionados.

CAPÍTULO 2

Redes Sociais Online

Antes de explorar as aplicações envolvendo processamento e visualização de dados de Redes Sociais Online (RSO), é necessário um claro entendimento do conceito de RSO, do histórico da área, funcionamento das principais plataformas e de como estas possibilitam a extração de seus dados para uso em aplicações externas. Assim, este capítulo apresenta com mais detalhes as definições acerca de redes sociais, um breve contexto histórico sobre o advento e popularização das principais RSO e como é possível extrair informação deste tipo de plataforma.

2.1 Definições e Histórico

Apesar de comumente usuários se referirem a redes sociais online como redes sociais apenas, a última denominação é mais abrangente. Redes sociais são definidas como estruturas compostas por atores sociais (tais como indivíduos ou organizações) e um conjunto de conexões entre estes, que representam algum tipo de interação (WASSERMAN; FAUST, 1994). No cotidiano, há diversos exemplos práticos de redes sociais: família, amigos, colegas de trabalho, entre outros. A estrutura que advém dessas relações geralmente se mostra complexa e a análise de redes sociais vem sendo um dos maiores paradigmas da sociologia contemporânea, sendo explorada também em várias outras ciências.

Elementos teóricos acerca do conceito de redes sociais datam da Grécia Antiga e, desde então, esforços têm sido investidos para se estudar relações sociais a partir dessas estruturas (COSTA et al., 2012). Contudo, foi só a partir da década de 1930 que foram observados estudos mais significativos na área, com o início do desenvolvimento de análises de interação por meio da construção de redes sociais que representam cenários reais. Estas análises permitem identificar padrões, localizar entidades de maior influência e outros aspectos sociais sobre a dinâmica deste tipo de estrutura.

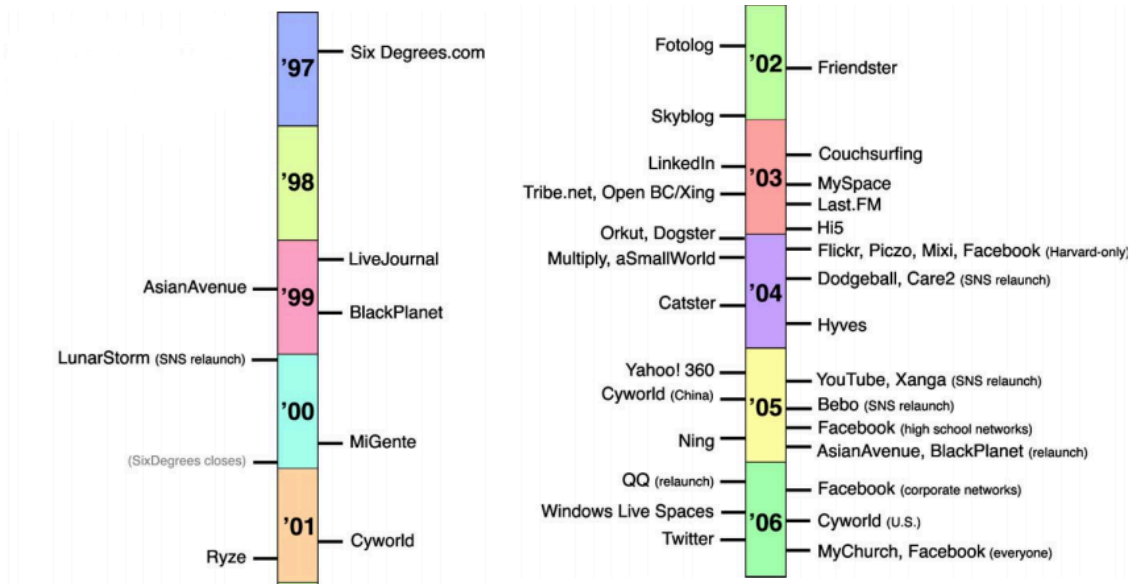
Com o surgimento da Internet e da *World Wide Web*, no fim da década de 1980, não demorou muito até que surgisse o conceito de Redes Sociais Online (RSO), que rapidamente ganharia popularidade. Formalmente, RSO são definidas como serviços web que permitem a indivíduos: (1) construir perfis públicos ou semipúblicos na rede; (2) articular uma lista de outros usuários da rede com os quais compartilham uma conexão; e (3) visualizar e acessar sua lista de conexões e as de outros criadas dentro do sistema. A natureza e o nome destas conexões varia de *website* para *website* (BOYD; ELLISON, 2007).

De acordo com a definição supracitada, a primeira RSO surgiu em 1997, chamada SixDegrees.com[6]. O serviço permitia a seus usuários criar perfis, listar seus amigos e navegar nas páginas destes amigos. Estas funcionalidades, olhadas separadamente, já existiam antes

[6] Deixou de funcionar em 2001 e atualmente não possui domínio registrado.

do surgimento da plataforma. Contudo, a SixDegrees foi a primeira a combiná-las num único serviço. A partir de então, várias outras RSO surgiram, especialmente a partir do início da década de 2000, com o lançamento de serviços mais populares, como o Friendster[7], MySpace[8], LinkedIn[9], entre outros. Apesar destas plataformas terem implementado diferentes funcionalidades, todas as suas estruturas básicas estavam de acordo com a definição dada anteriormente: consistiam em perfis que articulavam listas de amigos (outros usuários da plataforma). Em 2003, já havia dezenas de RSO em funcionamento, algumas com vários milhões de usuários registrados (DOS ANJOS; SAEGER; DA SILVA, 2013). A Figura 1 mostra uma linha do tempo com o surgimento das principais RSO até 2006.

Figura 1: Linha do tempo do surgimento das principais RSO até 2006



Fonte: Adaptado de BOYD e ELLISON (2007)

Algumas outras características, além da definição formal, também são normalmente compartilhadas por diferentes destes serviços. Tipicamente, após cadastrar-se numa RSO, o usuário é convidado a fornecer informações como idade, interesses, localização, entre outras, para que seu perfil possa ser construído a partir do conteúdo dado. A maioria das plataformas também permite e encoraja o usuário a publicar uma foto de perfil. Geralmente também é possível acrescentar conteúdo multimídia à página (como fotos adicionais ou vídeos), publicar textos e personalizar outros aspectos do perfil.

A capacidade de usuários personalizarem, articularem e visualizarem suas redes popularizou o conceito de redes sociais, que até então era utilizado primariamente por sociólogos para o estudo de relações interpessoais pré-existentes, distante do público geral. Tornou-se possível criar conexões entre indivíduos que não estão relacionados fora do contexto da RSO (apesar de normalmente este não ser o objetivo primário), algo que não faria sentido na definição original de redes sociais. Isso levanta uma importante característica

[7] https://www.friendster.com/

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

[9] https://www.linkedin.com/

das RSO: informações nelas representadas (atores sociais, conexões e outros tipos de informação) nem sempre condizem com a realidade e, tipicamente, não há nenhuma verificação da integridade de seus conteúdos.

Com o compartilhamento de informações pessoais na Web, um ponto que rapidamente ganhou atenção foi a privacidade dos usuários. A visibilidade das informações publicadas em RSO varia de plataforma para plataforma e possivelmente de acordo com configurações do usuário. Alguns serviços, como Friendster e Tribe.net[10], permitem indexação do conteúdo dos perfis de seus usuários em engenhos de busca e, portanto, tornam públicas essas informações. Outros, como Facebook[11] e Twitter[12] dão aos seus usuários controle sobre o acesso às suas informações, permitindo-os especificar quais conteúdos são públicos, disponíveis para contatos ou totalmente privados. Há também serviços que utilizam a privacidade do usuário como modelo de negócio ao disponibilizar a opção de controlar o acesso de suas informações apenas para membros pagantes (BOYD; ELLISON, 2007).

Usuários de RSO também têm a liberdade de especificar com quais usuários querem estabelecer uma conexão. O nome específico dado a esta conexão varia entre os serviços (amigo, contato, entre outros) e, normalmente, esta conexão é bidirecional, mas há exceções. O Twitter, por exemplo, utiliza conexões unidirecionais para construir o conceito de seguidor: um usuário A pode seguir um usuário B, sem que este precise segui-lo de volta.

Há várias outras características de RSO, além das apresentadas até então, que variam imensamente e as diferenciam entre si. Em termos de funcionalidades, algumas permitem compartilhamento de fotos e vídeos, outras dão suporte a mensagens privadas entre usuários, por exemplo. Em relação às plataformas utilizadas, algumas são restritas a dispositivos móveis ou ainda a sistemas operacionais específicos. Também há RSO com público-alvo definido, específicas para certas etnias, opções sexuais, visões políticas, entre outros tipos de categorização. Há, inclusive, RSO destinadas a animais, como a Dogster[13], para cachorros. Enfim, atualmente observa-se uma variedade imensa de RSO, cada uma com seus atrativos.

2.2 Cenário atual

Desde seu surgimento, as RSO têm atraído milhões de usuários e constituem atualmente a atividade online mais popular, à frente do e-mail pessoal e utilizada por mais de dois terços da população mundial online (BENEVENUTO; ALMEIDA; SILVA, 2011). O Facebook, por exemplo, atingiu em 2012 a marca de um bilhão de usuários registrados. Três anos mais tarde, em agosto de 2015, alcançou a marca de um bilhão de usuários diferentes conectados num único dia. Este crescimento espantoso tem sido particularmente acelerado devido à popularização de *smartphones* e *tablets* no mundo. Apenas em 2014, foram vendidos mais de 1,2 bilhões de *smartphones* no mundo (FRIEDMAN, 2015). Estes aparelhos facilitam o acesso à Internet e contribuem para a retenção dos usuários nas RSO.

[10] http://www.tribe.net/

[11] https://www.facebook.com/

[12] https://www.twitter.com/

[13] http://www.dogster.com/

A indiscutível popularidade alcançada por este tipo de serviço resultou numa mudança de paradigma para a disseminação de informação: mídias antes tratadas como fontes primárias de informação, tais como revistas ou jornais, têm perdido importância. Para Stempel (2000), o modelo de disseminação "muitos para muitos" está substituindo o modelo "um para muitos". Isto fica claro quando se observa a forma como a informação é compartilhada nas RSO atuais: vários usuários gerando e compartilhando conteúdo uns com os outros, levando a uma democratização na geração e divulgação de conteúdo (FRANÇA et al., 2014), que por muitos já é tratado como principal fonte de informação.

Tornou-se simples compartilhar informações de forma instantânea e com alto alcance. É raro, atualmente, encontrar um usuário da Internet que não seja membro de pelo menos uma RSO. A penetração destas plataformas é tão grande que vários serviços têm eliminado suas interfaces usuais de cadastro de usuários e substituído por botões que recuperam dados do usuário automaticamente de sua RSO preferida. Isto faz com que, cada vez mais, indivíduos sintam necessidade de cadastrar-se em plataformas sociais para possuírem uma identidade na Web.

Dentre as várias existentes, hoje, as cinco plataformas sociais mais populares são (em ordem decrescente de número de usuários ativos): Facebook, Twitter, LinkedIn, Pinterest e Google+, todas com mais de 100 milhões de usuários distintos ativos mensalmente[14]. Destas, apenas o Pinterest destina-se exclusivamente ao compartilhamento de fotos e vídeos e, por isso, não será abordado neste trabalho, já que este destina-se à análise do conteúdo textual disponível em RSO.

2.3 Extração de Dados

A larga quantidade de conteúdo criada por usuários de RSO rapidamente atraiu a atenção de cientistas de dados de grandes corporações que visualizaram o potencial destas plataformas para obter dados relevantes com facilidade. Para atender estas necessidades, RSO começaram a introduzir interfaces ou serviços com a finalidade de facilitar a captura, total ou parcial, de seu conteúdo por parte de desenvolvedores. A iniciativa começou com o Facebook, em 2007, com o lançamento de uma API para recuperação de dados públicos por meio de requisições HTTP GET[15]. A ideia foi rapidamente seguida por outras plataformas e atualmente a maioria das grandes RSO fornece algum facilitador para extração de seus dados.

Apesar de incomum, ainda há, contudo, casos de RSO que não provêm formas de acessar os dados de forma automática. Também é possível haver casos em que há uma API de consumo disponível, mas esta impõe limites indesejados, como restrição ao tipo de dado que pode ser recuperado, limite de quantidades, entre outros. Esses problemas não inviabilizam a extração de dados (apesar de acrescentarem algumas complicações): é possível desenvolver *crawlers*, que são aplicativos com o propósito específico de navegar por páginas

[14] Dado extraído do Alexa Global Traffic Rank, que ordena *websites* por popularidade. Disponível em <http://www.ebizmba.com/articles/social-networking-websites>

[15] Método do protocolo HTTP (*Hypertext Transfer Protocol*) para requisitar dados de um recurso específico através de uma URL.

Web para indexação ou recuperação e armazenamento de seu conteúdo. Neste contexto, pode-se utilizar este tipo de ferramenta para recuperar o conteúdo integral das páginas de uma RSO e filtrar a informação desejada.

A utilização de *crawlers* (sem API de consumo) é uma estratégia mais limitada, visto que depende de heurísticas baseadas no conhecimento prévio da forma como as páginas de uma determinada RSO são estruturadas. Assim, qualquer mudança na estrutura das páginas poderia fazer com que um *crawler* deixe de funcionar. Além disso, a recuperação de todo o conteúdo HTML[16] de uma página pode ser muito custoso. Do ponto de vista do tamanho do conteúdo transmitido na rede, é muito mais rápido transferir somente o conteúdo desejado do que uma página HTML inteira – por exemplo, para recuperar um post de 90 bytes no Twitter é necessário fazer o download de uma página de 870 kilobytes, tamanho quase 10 mil vezes maior, devido à inclusão de elementos de formatação e outros componentes não relevantes. Em termos de processamento, também há perdas, visto que muito tempo é gasto percorrendo todo o documento buscando as partes relevantes.

Devido aos problemas associados ao manuseio de conteúdo HTML original e à pressão dos desenvolvedores, a disponibilização de API de consumo tornou-se praticamente um componente obrigatório das grandes RSO. Hoje, das 5 plataformas sociais mais populares, citadas anteriormente, todas provêm API. As subseções seguintes detalham as características destas RSO e mostram como funciona o processo de extração de dados por meio de suas API (com exceção do Pinterest[17], por tratar de imagens e vídeos, como já citado).

2.3.1 Twitter

Criado em 2006, o Twitter é um serviço de *microblog* em tempo real que permite ao usuário publicar em sua página *tweets*, mensagens curtas (com limite de 140 caracteres) que normalmente correspondem aos pensamentos, ideias ou o que está acontecendo ao redor do usuário. Dessa forma, o Twitter pode ser visto como uma infraestrutura com foco em prover comunicação rápida e fácil, permitindo aos seus usuários consumir muita informação com velocidade. O serviço ganhou popularidade rapidamente e conta, hoje, com mais de 500 milhões de usuários, dos quais mais de 300 milhões são ativos.

Usuários do Twitter conectam-se seguindo uns aos outros. Ao seguir um usuário, novos *tweets* do mesmo serão mostrados na página inicial. É importante notar que, ao contrário da maioria das RSO, as conexões entre usuários no Twitter são assimétricas. Ou seja, é possível (e bastante comum) que um usuário siga alguém que não o segue de volta. Usuários podem se comunicar por meio de *tweets* ou mensagens diretas. No primeiro, basta mencionar o destinatário (escrevendo o caractere '@' seguido do nome do usuário em questão) e este receberá uma notificação de que foi mencionado em um *tweet*. A segunda permite que usuários enviem mensagens privadas uns aos outros, desde que conectados.

---

16 *HyperText Markup Language*, linguagem de marcação interpretada por navegadores e utilizada na produção de páginas na Web.

17 https://www.pinterest.com/

Dados do Twitter são particularmente relevantes porque são, mais do que em qualquer outra rede social, compartilhados quase à mesma velocidade dos pensamentos e imediatamente disponíveis para consumo (RUSSELL, 2013). Dentro do escopo de análise de sentimentos e mineração de opinião, é seguro dizer que o Twitter é a fonte de dados mais apropriada para esta finalidade, visto que é a RSO mais utilizada quando se pensa em expressar opiniões e narrar experiências.

Além de ser uma fonte rica de dados sociais, o Twitter provê duas API, limpas e bem documentadas, para desenvolvedores cadastrados: REST API[18] e *Streaming* API[19]. A primeira foca no suporte a escrita e leitura programática de *tweets* (busca por conteúdo e publicação de novos *tweets*), enquanto a segunda provê monitoramento praticamente em tempo real de *tweets* (TWITTER, 2015).

Ao cadastrar-se como desenvolvedor, o Twitter disponibiliza credenciais utilizadas para autenticar aplicações a fim de garantir acesso aos seus dados.. A Figura 2 mostra um exemplo simples, implementado em Python 2.7[20], de utilização da REST API para recuperar e imprimir os 10 *tweets* mais recentes utilizando a consulta "CIn UFPE" (incluindo todos que possuam as palavras "CIn" e "UFPE", não necessariamente consecutivas no texto) e parte da saída gerada. Este exemplo ilustra, em poucas linhas, a facilidade de uso da API, cujo funcionamento é extensivamente abordado em sua documentação.

Figura 2: Exemplo de extração de tweets utilizando API do Twitter e parte do conteúdo recuperado

```
import twitter
import json

CONSUMER_KEY = [illegible]
CONSUMER_SECRET = [illegible]
OAUTH_TOKEN = [illegible]
OAUTH_TOKEN_SECRET = [illegible]

auth = twitter.oauth.OAuth(OAUTH_TOKEN, OAUTH_TOKEN_SECRET,
 CONSUMER_KEY, CONSUMER_SECRET)
twitter_api = twitter.Twitter(auth=auth)

resultados_busca = twitter_api.search.tweets(q='CIn UFPE', count=10)
tweets = resultados_busca['statuses']

for tweet in tweets:
 print '@%s: %s\n' % (tweet['user']['screen_name'], tweet['text'])
```

```
@CInUFPE: Aluno e ex-alunos do CIn-UFPE criam ferramenta que formata textos
normas da ABNT. Saiba mais: http://t.co/325E2QplHa

@grumpylims: netflix and chill #sofanocin @ CIn - Centro de Informática (UFP
t.co/OSq3KUB7dc

@gustavohfilho: I'm at CIn - Centro de Informática da UFPE - @cinufpe in Rec
ps://t.co/PgV1fCIBT3

@littleamanda_: Hell vulgo CIn, UFPE

@miguelarauj1o: Obrigado Pai por mais um dia. @ CIn - Centro de Informátic
tps://t.co/KB753cfx2p
```

Fonte: O Autor

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

[19] https://dev.twitter.com/streaming/public

[20] https://www.python.org/

As API do Twitter utilizam o padrão JSON[21] para serialização do conteúdo recuperado, que inclui, além do texto propriamente dito contido no *tweet*, vários outros metadados, como URL[22] contidas, *hashtags* (elementos utilizados para categorização do conteúdo publicado), mídia, localização, entre outros. Além de recuperação de *tweets*, a REST API disponibiliza praticamente todo o conteúdo público do Twitter de forma programática. Esta quantidade imensa de informação facilmente acessível faz da REST API uma das ferramentas mais poderosas para extração de dados sociais.

2.3.2 Facebook

Fundado em 2004, o Facebook é uma RSO gratuita na qual usuários administram perfis, onde podem publicar posts (textos ou conteúdo multimídia), listar seus interesses e conectar-se com outros usuários (conexões simétricas). Usuários têm a liberdade de configurar quem pode ter acesso ao conteúdo de seus perfis, aceitar ou recusar pedidos de conexão e interagir com páginas de outros tipos de entidades, tais como restaurantes, universidades, escolas, entre outros. Hoje, a plataforma conta com mais de 1,2 bilhão de usuários ativos, dos quais mais de 60% utiliza a RSO diariamente.

O Facebook disponibiliza a Graph API[23] para acesso parcial aos seus dados por meio de requisições HTTP GET. Para utilizá-la, é preciso cadastrar-se como desenvolvedor em uma conta pré-existente na plataforma e criar uma aplicação para que, então, se gere uma *access token* que será utilizada para autenticar o acesso aos dados. O conteúdo recuperado, assim como no Twitter, é serializado no formato JSON. A Figura 3 mostra um exemplo, implementado em Python 2.7, de utilização da Graph API e parte do conteúdo recuperado.

Figura 3: Exemplo de utilização da Graph API do Facebook

```
1   import requests
2   import json
3
4   ACCESS_TOKEN = """CAACEdEose0cBADLr9LBooc93oXlcCwZApEhTNoDHGU3iu9IeFWIw9F6MMkXe
5   HL3ehnFyvc8gB0COkxOFpRxsaJJpqjT90ynTnBOiEYyuoizYUdWcISXAdCXXAUbZCfN1dw43ecJBsHK
6   SZBM8lxlUbVCuHEsT77xrsdlE1Bf9td0jCYQZCSQkqgTqBuVoGJxlphLabDxZChbKN5JxLb1l0"""
7
8   q = 'CIn UFPE'
9
10  url_base = "https://graph.facebook.com/"
11  url_consulta = "search?q="+str(q)+"&type=page&access_token="+str(ACCESS_TOKEN)
12
13  content = requests.get(url_base+url_consulta).json()
14  print json.dumps(content, indent=1)
```

```
"data": [
 {
  "category": "University",
  "name": "CIn UFPE Oficial",
  "category_list": [
   {
    "id": "108051929285833",
    "name": "College & University"
```

Fonte: O Autor

[21] *JavaScript Object Notation*, formato para intercâmbio de dados computacionais.

[22] *Uniform Resource Locator*, padrão global de endereço para recursos na Web.

[23] https://developers.facebook.com/docs/graph-api/

O exemplo de Figura 3 mostrou a recuperação de páginas a partir da consulta "CIn UFPE". Além de páginas, a Graph API permite ao desenvolvedor realizar buscas por diversas entidades (e metadados associados) que contenham um ou mais termos dados como parâmetros na consulta. Estas entidades incluem usuários do Facebook, grupos de usuários, eventos, entre outros. Contudo, desde 2014, o Facebook removeu a busca por *posts* públicos (FACEBOOK, 2015). Esta remoção compromete fortemente a utilização da API do Facebook para aplicações de análise de sentimentos, já que são nos *posts* que usuários tipicamente expressam suas opiniões.

2.3.3 LinkedIn

O LinkedIn é uma RSO focada em negócios e possui como público alvo profissionais em geral. Foi fundada em 2002 e permite aos seus membros estabeleceram conexões entre si, recomendar profissionais e suas habilidades, navegar por oportunidades de emprego, interagir diretamente com empregadores e personalizarem seus perfis (focam em histórico profissional, educação, certificações, entre outros). Uma diferença do LinkedIn em relação às outras RSO citadas até então é a exigência de que dois usuários se conheçam de alguma forma para estabelecerem uma conexão, sendo preciso especificar como os usuários se conheceram.

Além da comunicação de usuários por meio de mensagens privadas, membros do LinkedIn também podem compartilhar conteúdo utilizando publicações (texto ou conteúdo multimídia). Nas páginas inicias dos usuários são mostradas publicações dos usuários conectados e alterações recentes em seus perfis (semelhante ao funcionamento do Facebook). Também é possível seguir empresas (conexão unilateral, diferentemente da conexão entre membros) para que publicações dessas empresas sejam incluídas em sua página inicial.

Assim como o Twitter e no Facebook, o LinkedIn provê API que permitem acesso programático às suas informações. Duas são particularmente populares: Connections API e Search API. A primeira permite acesso à lista de conexões do usuário e informações básicas dos perfis destas conexões. A segunda permite consultas (por meio de termos) para recuperar pessoas, empresas, empregos ou publicações disponíveis no LinkedIn. Para obter as chaves de acesso às API disponíveis, é necessário possuir uma conta na plataforma e identificar-se como desenvolvedor criando uma aplicação.

É possível extrair diversos tipos de dados relevantes do LinkedIn, que permitem análises interessantes, como investigar os locais de trabalho mais populares entre ex-alunos de uma universidade, cidades com mais empregos em uma determinada área, entre outros. No contexto de mineração de opinião, o conteúdo de maior relevância são as publicações do usuários. Contudo, devido ao caráter corporativo da RSO, esta raramente é usada por seus membros para expressar opiniões. A grande maioria das publicações é desprovida de sentimentos e foca em informações profissionais e oportunidades de emprego. Dessa forma, apesar da popularidade do LinkedIn em termos de usuários ativos e existência de uma API bastante completa (cuja utilização é ilustrada na Figura 4), a extração de seus dados para aplicações de análise de sentimentos é pouco relevante (apesar de possível).

Figura 4: Exemplo de utilização da API do LinkedIn e parte do conteúdo recuperado

```
1  import requests
2  import json
3
4  ACCESS_TOKEN = "[redacted]"
5  [redacted]
6  [redacted]
7
8  keywords = "UFPE"
9  url_base = "https://api.linkedin.com/v1/"
10 url_consulta = ("company-search:(companies:(id,name,website-url,industries,"
11                 "status,locations,description,founded-year,num-followers))?keywords==")
12 params_consulta = "&oauth2_access_token=" + ACCESS_TOKEN + "&format=json"
13
14 content = requests.get(url_base+url_consulta+keywords+params_consulta).json()
15 print json.dumps(content, indent=1)
```

```
 "values": [
  {
   "code": "68",
   "name": "Higher Education"
  }
 ]
},
"websiteUrl": "www.ufpe.br",
"status": {
 "code": "OPR",
 "name": "Operating"
},
"id": 28514,
"name": "UFPE",
```

Fonte: O Autor

2.3.4 Google+

Após algumas tentativas falhas, o Google lançou em 2011 o Google+ como sua quarta RSO com a intenção de competir com outras gigantes já estabelecidas. A plataforma combina serviços do Twitter e Facebook em uma única RSO, com a intenção de atrair usuários dos dois rivais (RUBEN CUEVAS et al., 2013).

No Google+, cada usuário possui um perfil, no qual são mostradas suas atividades e momentos, como no Facebook. Momentos consistem em atualizações no perfil do usuário (e.g. começou um relacionamento sério, mudou-se de cidade, entre outros). Atualizações são publicações que consistem em textos de tamanhos variáveis e possivelmente anexos, que podem ser fotos, vídeos ou qualquer outro arquivo. Conexões entre usuários são unidirecionais, e um usuário A conectado a B (chamado de seguidor de B) recebe atualizações de B em sua página inicial (exatamente como no Twitter).

Como nas outras RSO, usuários do Google+ podem cadastrar uma aplicação para obter chaves de acesso à sua API de consumo. Com as chaves, é possível enviar requisições HTTP GET para recuperar conteúdos serializados em JSON. A API do Google+ é bastante completa, além de bem documentada, permitindo extração vários tipos de dados, como informações de usuários, atividades públicas, comentários de atividades e momentos de usuários específicos (GOOGLE, 2015). A Figura 5 mostra um exemplo, implementado em Python 2.7, de extração de atividades (publicações) públicas do Google+.

Figura 5: Exemplo de utilização da API do Google+ e parte do conteúdo recuperado

```
import httplib2
import json
import apiclient.discovery

API_KEY = 
service = apiclient.discovery.build('plus', 'v1', http=httplib2.Http(), developerKey=API_KEY)

atividades_recurso = service.activities()
atividades = atividades_recurso.search( \
    maxResults=5,orderBy='best',query='CIn UFPE').execute()

if 'items' in atividades:
  for atividade in atividades['items']:
    print atividade['actor']['displayName'], ': ', atividade['object']['content'], '\n'
```

```
Leonardo Andrade :  Mais um dia no CIn UFPE, que alegria! Eu amo esse lugar!

Duhan Caraciolo Maia Souza :  Quase quatro anos de aulas no CIn UFPE.

Rafael Gouveia :  Esse CIn  é  muito maroto aqui na UFPE.

Openredu :  CIn-UFPE recebe palestra sobre Modelo de Negócios para Bens Intangíveis e Software
livre. <br /><br />Palestrante: Corinto Meffe, assessor da presidência do SERPRO<br />Local: An
fiteatro do CIn-UFPE<br />Data/Horário: 28/03/14 (sexta-feira), às 9h<br /> <br />Mais informaç
ões: <a href="http://www2.cin.ufpe.br/site/lerNoticia.php?s=1&amp;c=94&amp;id=908">http://www2.
cin.ufpe.br/site/lerNoticia.php?s=1&amp;c=94&amp;id=908</a>

Andre Rincon :  CIn-UFPE abre inscrições para duas turmas do Curso de Residência em Software<br
 /><br />Interessados têm até o dia 25 de agosto para se inscrever na seleção<br /><br />Em par
ceria com a Motorola, o Centro de Informática (CIn) da UFPE abre inscrições para a décima quint
```

Fonte: O Autor

No exemplo acima, foi utilizada a consulta “CIn UFPE” para recuperar cinco publicações mais recentes. O resultado é similar ao recuperado do Twitter e inclui *posts* com conteúdos variados. A facilidade de uso de sua API e possibilidade de buscar publicações públicas de usuários a partir de termos específicos faz do Google+ uma excelente fonte de dados para diversos tipos de aplicações, inclusive análise de sentimentos.

2.4 Considerações Finais

Levando em consideração a penetração das RSO no mundo ao longo dos últimos anos e o volume de conteúdo gerado e compartilhado nestas plataformas por um número cada vez maior de usuários, fica claro que os dados oriundos deste tipo de serviço são bastante valiosos. Ao se tratar de um domínio de aplicação bem definido, como análise de sentimentos, alguns cuidados na escolha das plataformas são necessários para garantir a relevância dos dados utilizados.

A partir da análise das API das principais RSO, é possível observar que algumas plataformas destacam-se como alternativas mais viáveis para análise de sentimentos, ao passo que outras apresentam algumas limitações que as tornam menos apropriadas para este fim. O Facebook e o LinkedIn, apesar de muito populares, perdem destaque dentro deste escopo. A primeira, por ter removido a busca por *posts* públicos, tanto de sua API como do serviço web, inviabilizando a extração de opiniões de usuários. Já o LinkedIn, devido ao seu caráter fortemente corporativo e tipicamente desprovido de opiniões.

O Twitter e o Google+, dentre as RSO analisadas, mostram-se excelentes fontes de dados sociais para mineração de opinião. Devido ao maior número de usuários e por ser amplamente considerado o serviço mais usado para publicação de opiniões, o Twitter foi escolhido para ilustrar o funcionamento de uma arquitetura baseada em análise de sentimentos e *Business Intelligence*, que será mostrada em detalhes no Capítulo 5 deste trabalho. Num cenário real de utilização, contudo, caberia explorar várias outras plataformas não citadas neste trabalho, mas que seguem modelos semelhantes de disponibilização de seus dados.

Após o estudo do processo de extração de dados sociais, é necessário avaliar formas de processar este tipo de informação automaticamente para que se possam ser inferidas informações relevantes para organizações. Assim, o capítulo seguinte deste trabalho irá explorar formas de classificar conteúdo de RSO a partir das polaridades das opiniões expressas (análise de sentimentos).

CAPÍTULO 3

Análise de Sentimentos

Devido à popularização de redes sociais online, atualmente a Web conta com uma quantidade imensa de conteúdo carregado de opinião. Com isso, surgem oportunidades para explorar tecnologias que sejam capazes de procurar e processar opiniões na Internet. Esta motivação fez com que a área de análise de sentimentos (também chamada mineração de opinião), a qual lida com o tratamento computacional e classificação de sentimentos expressos em textos, ganhasse ainda mais atenção.

Este capítulo mostrará como é possível classificar fragmentos de texto de acordo com a polaridade da opinião nele expressa. Inicialmente, é mostrada uma visão geral da área e possíveis abordagens. Em seguida, são abordadas técnicas para preparação do conteúdo social que normalmente são utilizadas para que então sejam explorados alguns classificadores conhecidos na literatura. O capítulo é encerrado apresentando implementações dos classificadores mostrados e comparações de desempenho a partir de dados extraídos de redes sociais online.

3.1 Visão Geral

Extrair sentimentos automaticamente de plataformas sociais pode ser extremamente útil por permitir análises sobre diversos temas sem intervenção manual. É claro o interesse por parte de organizações, as quais podem buscar opiniões públicas sobre suas marcas ou produtos e analisar satisfação de seus clientes. Também pode haver interesse por parte de consumidores, já que estes podem utilizar ferramentas de análise de sentimentos para pesquisar produtos ou serviços antes de realizar compras (GO et al., 2009).

Nos últimos anos, especialmente a partir do início da década dos anos 2000, devido à popularização da *World Wide Web*, houve muita atividade de pesquisa na área de classificação de sentimentos. A maior parte destas pesquisas inicialmente focou em classificação de quantidades largas de texto, como resenhas de produtos, por exemplo (PANG; LEE, 2008). Dados sociais são um pouco diferentes principalmente devido a sua casualidade e tamanhos relativamente menores. Contudo, com a larga utilização de RSO, várias pesquisas têm tratado especificamente de análise de sentimentos dentro do contexto de plataformas sociais, com destaque para o Twitter[24].

Apesar de vários autores se referirem à análise de sentimentos ao tratar de análises mais complexas e de extração de sentimentos, o termo normalmente diz respeito à classificação de fragmentos de texto com relação à sua polaridade apenas. Ou seja, ao tratar de mineração de opinião, o objetivo principal é classificar padrões em uma dentre três possíveis classes: (1) positivo, quando o texto revela emoções como felicidade, aprovação ou

[24] https://www.twitter.com/

divertimento; (2) negativo, quando exprime tristeza, raiva, desapontamento ou similares; e (3) neutro, quando simplesmente constata um fato com imparcialidade ou não expressa emoção. O Quadro 1 mostra um exemplo de 3 *tweets* extraídos do Twitter e suas respectivas classificações.

Quadro 1: Exemplo de classificação de *tweets*

Tweet	Classificação
@larissapassos: Só vi 3 filmes do Oscar esse ano, mas dos que vi, Birdman foi o melhor disparado	Positivo
@TheGhostSix: Ano passado a Fox Searchlight colocou Grand Budapest e Birdman no Oscar.	Neutro
@carolresendee: Birdman, pior filme que já vi, como que ganhou?	Negativo

Fonte: O Autor

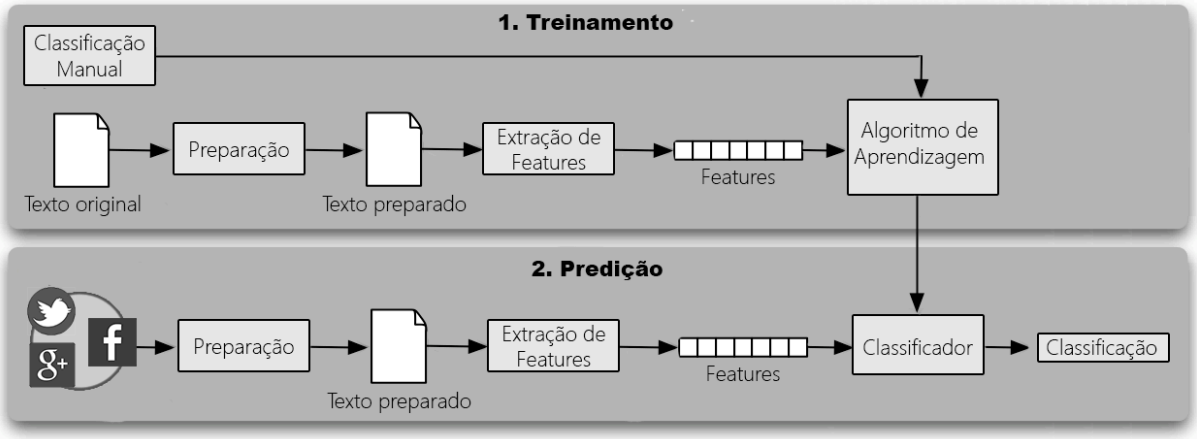
Tipicamente, duas abordagens podem ser utilizadas para classificar textos quanto à sua polaridade: abordagem simbólica ou aprendizagem de máquina. Na primeira, um texto é considerado uma coleção de palavras, geralmente sem relação entre si. Especialistas humanos são encarregados de definir o grau de polaridade de uma coleção pré-definida de palavras e, então, a polaridade de um fragmento de texto pode ser calculada por meio da agregação dos sentimentos das palavras que o compõem utilizando alguma função matemática como média ou somatório (BOIY; MOENS, 2009).

Técnicas de aprendizagem de máquina constroem modelos para classificação a partir de um conjunto de treinamento e um algoritmo de aprendizagem. PANG et al. (2002) mostraram que abordagens simbólicas são fortemente dependentes de contexto (por exemplo, o conjunto de palavras com polaridade positiva para o contexto de resenha de filmes pode não ser útil para discussões sobre racismo). Apesar de também haver este tipo de dependência tratando-se de aprendizagem de máquina, esta abordagem tem, tipicamente, alcançado melhores resultados em aplicações que não possuem domínio específico, principalmente devido a sua capacidade de capturar contexto e ser facilmente adaptável.

De forma geral, os modelos de classificação utilizados para análise de sentimentos não recebem o fragmento de texto original como entrada. Em vez disso, o texto é convertido numa coleção de *features* ou características, que são normalmente segmentos do texto (originais ou modificados) capazes de representar seu conteúdo, associados a pesos. Para que os classificadores tenham um bom desempenho, é imprescindível uma boa construção de um vetor de *features* representativo.

Técnicas de aprendizagem de máquina são divididas entre supervisionadas, quando o treinamento se dá com exemplos previamente classificados, e não supervisionadas, quando não há categorização prévia no treinamento. Atualmente, métodos supervisionados são os mais utilizados atualmente em mineração de opinião (BOIY; MOENS, 2009). A Figura 6 esquematiza uma arquitetura típica baseada em classificadores supervisionados para análise de sentimentos de dados sociais.

Figura 6: Arquitetura típica de análise de sentimentos de conteúdo social baseada em aprendizagem de máquina



Fonte: O Autor

Conforme ilustrado na Figura 6, além da extração de *features* propriamente dita, normalmente também é realizada uma etapa de preparação do conteúdo antes da aprendizagem ou classificação. As seções seguintes explorarão com maior profundidade as fases esquematizadas na figura e, por fim, são mostradas implementações de diferentes classificadores e comparações de seus desempenhos.

3.2 Preparação do Conteúdo e Extração de *Features*

Antes de considerar diferentes classificadores para analisar sentimentos em textos, é necessário preparar o conteúdo para eliminar partes não relevantes e possivelmente adaptar trechos para otimizar o desempenho dos classificadores. Esta preparação envolve a utilização de várias técnicas de pré-processamento que têm sido exploradas na grande área de recuperação de informação nas últimas décadas. O resultado final deve ser um vetor de *features* capaz de representar o texto original e que possa ser utilizado como entrada de um classificador.

3.2.1 Filtro de *stopwords*

Em fragmentos de texto, nem todas as palavras são úteis para representar a semântica do documento. Por exemplo, artigos, preposições ou conjunções não carregam nenhum significado e, portanto, podem ser ignorados. Estes tipos de construções gramaticais, consideradas inúteis em termos de opinião, são chamados de *stopwords* e normalmente são eliminadas do texto. Devido à sua ampla utilização em abordagens de classificação de texto em geral, atualmente há várias listas públicas de *stopwords* disponíveis para os mais diversos idiomas que podem ser utilizadas em diferentes aplicações.

A eliminação de *stopwords*, além de melhorar o desempenho de classificadores, tem um benefício adicional: reduz o tamanho do texto significativamente. Tipicamente se obtém

uma compressão de 40% ou mais no tamanho do texto apenas eliminado *stopwords* (BAEZA-YATES, 2011). Essa redução no tamanho do texto traz consigo ganhos perceptíveis em termos de tempo gasto no treinamento e na classificação dos algoritmos.

3.2.2 Aplicação de *stemming*

É comum a utilização de variantes de palavras na construção de textos. Plurais ou gerúndios são exemplos de variações sintáticas que podem ser utilizadas e que não têm distinção de polaridade em relação às suas formas originais (BAEZA-YATES, 2011). Por exemplo, as palavras incrível, incríveis e incrivelmente, apesar de diferentes, de forma geral representam sentimentos positivos. Dessa forma, fragmentos de texto podem ser simplificados por meio da substituição dessas variações pelos seus *stems*. Estes referem-se à porção de uma palavra obtida após remoção de seus afixos (prefixos e sufixos). Este processo de remoção de afixos é normalmente chamado de *stemming*.

Apesar da aplicação de *stemming* ser mais comum em aplicações de indexação de conteúdo, pois permite casamentos entre diferentes variações de uma mesma palavra, alguns autores argumentam que esta operação também pode ser útil dentro do contexto de análise de sentimentos, visto que classificadores podem tratar variantes de uma mesma palavra da mesma forma, aumentando a probabilidade de que um classificador já tenha sido treinado para estas variantes e simplificando o processo de aprendizado. Contudo, também argumenta-se que, ao reduzir variantes a *stems*, é comum perder informação sobre a polaridade de palavras e, portanto, a eficácia de algoritmos de *stemming* em aplicações de mineração de opinião é questionável (POTTS, 2011).

3.2.3 Filtro de caracteres especiais

Alguns caracteres, assim como palavras (*stopwords*, discutidas anteriormente), também não acrescentam semântica para fragmentos de texto. É o caso de caracteres de pontuação (pontos finais, de interrogação ou exclamação, vírgulas, aspas, entre outros). Alguns cuidados são necessários na implementação de filtros de pontuação, pois é preciso diferenciá-los de *emoticons*, que são combinações de caracteres utilizadas para simular expressões faciais (“:)”, “:(“, “:D”, entre outros) e consistem num conteúdo extremamente rico em termos de polaridade de sentimentos (KUNNEMAN et al., 2014).

Além de pontuação, tratando-se de conteúdo social, é normal usuários escreverem de maneira informal e até propositalmente incorreta. Por exemplo, é bastante comum repetir letras de uma mesma palavra para enfatizar suas emoções: escrever “ameeeeei” ou “ameiiii” em vez de “amei” é um exemplo de repetição proposital de caracteres. Assim, é indicado tratar estes casos no pré-processamento do texto (com o cuidado de respeitar a ortografia do idioma em questão – neste caso, um tratamento específico por idioma se faz necessário).

3.2.4 Extração de *features*

A última etapa de preparação do conteúdo social é a extração de *features*[25], que consiste na segmentação do texto em *tokens* (elementos representando características) e pesos associados. Há pesquisas que tratam especificamente de extração de *features* para abordagens de aprendizagem de máquina em geral, bem como abordagens para problemas específicos de categorização de textos e recuperação de informação (PANG; LEE; VAITHYANATHAN, 2002). Assim, diferentes abordagens para extração de *features* têm desempenhos diferentes a depender do problema a ser resolvido.

A forma mais comum de extração de *features* é por meio da extração de *unigrams*, que são simplesmente palavras (ou seja, extração de conjuntos unitários de palavras que ocorrem no texto). Apesar de esta abordagem ser amplamente utilizada e aplicável para praticamente qualquer tipo de texto, limitações podem ser facilmente identificadas. Alguns exemplos são mostrados no Quadro 2. No primeiro caso, ao separar a construção "não gostei" em duas *features* independentes, "gostei" pode erroneamente acrescentar uma polaridade positiva a uma mensagem que claramente expressa uma opinião negativa. Já no segundo, um problema parecido acontece ao separar "incrivelmente chato" e gerar uma *feature* "incrivelmente".

Quadro 2: Exemplo de problemas na utilização de *unigrams*

Tweet	Features
@MariaGueeedes: Não gostei foi da música de Teló kkkkk #TheVoiceBrasil	"não", "**gostei**", "foi", "música", "teló", "kkkkk"
@yasm3llo_: Nunca assisti todos os filmes de Bridget Jones. O primeiro é tão incrivelmente chato ZzzzZZzzzZZz	"nunca", "assisti", "todos", "filmes", "bridget", "jones", "primeiro", "é", "tão", "**incrivelmente**", "chato"

Fonte: O Autor

Uma abordagem que busca solucionar os problemas levantados é a utilização de *bigrams* ou *n-grams*, que são, respectivamente, conjuntos binários e n-ários de palavras como *features*. Neste caso, "não gostei" ou "incrivelmente chato" seriam preservados. Contudo, esta abordagem traz consigo um novo problema: o espaço de *features* torna-se muito esparso para n maior que 1 e a acurácia geral do sistema não é melhorada (GO et al., 2009). Por esta razão, normalmente *bigrams* não são utilizados (puramente) para análise de sentimentos. Alguns autores optam por combinar a utilização de *unigrams* e *bigrams*, já outros utilizam apenas *unigrams.* PANG; LEE; VAITHYANATHAN (2002) realizaram um comparativo e mostraram que a utilização de *unigrams* obteve melhores resultados quando comparada à utilização de *n-grams* ou abordagens híbridas.

[25] É importante ressaltar que a denominação *features* também pode referir-se às entidades que podem ser analisadas para avaliação de um determinado produto ou serviço, no contexto de mineração de opinião. Contudo, neste trabalho, *features* refere-se aos elementos que representam um fragmento textual, definição amplamente utilizada em trabalhos de classificação de texto.

É comum associar pesos às *features* coletadas para indicar para o classificador a importância de cada um em relação aos outros. A forma mais simples de representação é a indicação da frequência, isto é, o número de vezes que cada *feature* ocorreu no texto. Uma forma similar é a indicação de presença que, em vez de contar o número de ocorrências, utiliza apenas um booleano que indica se o *feature* ocorreu (uma ou mais vezes) ou não no texto, sem distinção entre *features* que ocorreram uma única vez ou múltiplas vezes.

Existem outras abordagens populares para associação de pesos utilizadas em recuperação de informação, como o TF-IDF, que calcula o peso a partir da frequência e da frequência inversa, que é o quociente do número total de documentos e o número de documentos no qual o termo ou *feature* aparece. Apesar de amplamente utilizadas em outros escopos, estas abordagens têm sido pouco utilizadas para mineração de opinião. PANG et al. (2002) realizaram o primeiro trabalho a comparar estas diferentes abordagens especificamente para análise de sentimentos, e mostraram que melhores resultados foram obtidos utilizando a indicação da presença da *feature* apenas, sem contagem de múltiplas ocorrências. Por esta razão, os testes realizados no fim deste capítulo restringem-se à indicação de presença das *features*.

3.3 Classificadores

Após a preparação do conteúdo e extração de *features*, o próximo estágio numa arquitetura baseada em aprendizagem de máquina é a aprendizagem (para o caso do treinamento) ou a classificação propriamente dita. Para isso, se utiliza um algoritmo de aprendizagem capaz de aprender com um conjunto de treinamento (normalmente dados manualmente classificados) e construir um modelo de predição baseado nestes dados.

Existem diversos algoritmos na literatura com o propósito de classificar texto. Neste trabalho, será apresentado o funcionamento geral de três dentre os mais populares classificadores para a análise de sentimentos. Para um estudo mais aprofundado sobre o funcionamento de cada um deles e a teoria matemática que os embasa, recomenda-se consultar outras referências que tratem especificamente deste tópico.

3.3.1 Bayesiano Ingênuo

O classificador Bayesiano ingênuo (LEWIS, 1998) é um dos mais simples classificadores probabilísticos e tem a regra de Bayes[26] como equação principal. É assim chamado por assumir (ingenuamente) que cada *feature* individualmente contribui para uma das classes de forma independente uma das outras (PANG et al., 2002). Formalmente, a probabilidade de um documento d pertencer a uma classe c, segundo a regra de Bayes, é:

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

[26] Regra que mostra a relação entre uma probabilidade condicional e sua inversa, criada pelo matemático inglês Thomas Bayes.

O modelo é construído a partir da análise da distribuição das ocorrências de cada *feature* nos textos do conjunto de treinamento para estimar $P(d|c)$. Este termo é reescrito como sendo o produto da probabilidade de ocorrência de cada das m *features* do documento d na classe c, $P(f_i|c)$, estimada como o número de indivíduos da classe c que têm ocorrências de f_i dentre o número total de indivíduos da classe. Assim, a regra pode ser adaptada para:

$$P(c|d) = \frac{P(c)(\prod_{i=1}^{m} P(f_i|c))}{P(d)}$$

O classificador é bastante simples e, tipicamente, há dependência condicional entre *features* em problemas reais (ignoradas em sua classificação). Contudo, ainda assim, classificadores Bayesianos ingênuos têm sido amplamente utilizados em problemas do mundo real e atingido taxas de acerto surpreendentemente altas (PANG et al., 2002).

3.3.2 Entropia Máxima

Um segundo classificador popular na literatura é o de entropia máxima. O classificador opera sob o princípio de preservação do máximo de incerteza possível: quando não se sabe nada, deve-se manter a distribuição o mais uniforme possível, ou seja, com entropia máxima (NIGAM et al., 2009).

O conjunto de treinamento é utilizado para estimar probabilidades condicionais a partir da análise do comportamento dos dados, inferindo restrições para o modelo que caracterizam as expectativas para a distribuição. Então, um algoritmo iterativo é utilizado para encontrar a distribuição de entropia máxima que seja consistente com as restrições aprendidas a partir do conjunto de treinamento.

No contexto de análise de sentimentos ou de classificação de texto em geral, as restrições são construídas a partir do cálculo do valor esperado das *features* no conjunto de treinamento. Por exemplo, caso 80% dos documentos no conjunto de treinamento em que a *feature* "excelente" possuam classificação positiva, o modelo construirá uma restrição que expresse esse comportamento. Assim, cada restrição significa uma característica do conjunto de treinamento que deve estar presente na distribuição aprendida. DELLA PIETRA; LAFFERTY (1997) mostraram que, quando as restrições são assim estimadas, é garantido que há uma distribuição única de máxima entropia que as satisfaça, da forma exponencial:

$$P(c|d) = \frac{1}{Z(d)} \exp\left(\sum_i \lambda_i f_i(d,c)\right) \quad (1)$$

$$Z(d) = \sum_c exp\left(\sum_i \lambda_i f_i(d,c)\right) \quad (2)$$

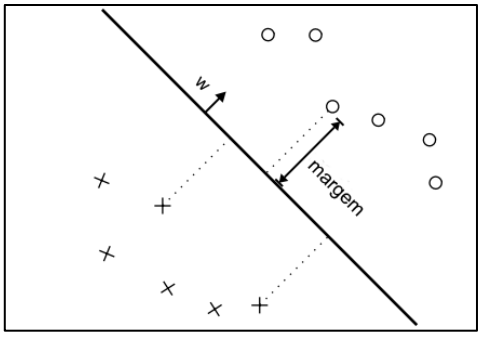
Na equação (1), $f_i(d,c)$ representa uma *feature,* enquanto $Z(d)$ é um fator de normalização, calculado em (2). λ representa o vetor de pesos das *features,* a ser estimado por meio de otimização. É garantido que a superfície de solução para o problema de entropia

máxima é convexa, com um único máximo global e sem máximos locais. Assim, uma abordagem comumente utilizada é partir de uma distribuição exponencial aleatória e utilizar um algoritmo de subida da encosta para convergir para o máximo global (NIGAM et al., 2009).

3.3.3 Máquina de Vetores de Suporte

Máquina de vetores de suporte (comumente chamada de SVM, sigla para *Support Vector Machine*), diferentemente dos classificadores mostrados até então (probabilísticos), trata-se de um classificador de margem máxima. Seu funcionamento se dá por meio do mapeamento todos os indivíduos no espaço n-dimensional (n é o número de *feature*s) e construção de um hiperplano que não somente separe as classes, mas que também possua distância Euclidiana máxima dos exemplos de treinamento pertencentes a classes opostas mais próximos. Este hiperplano é chamado de hiperplano de margem máxima e um exemplo para o caso bidimensional é mostrado na Figura 7.

Figura 7: Exemplo de hiperplano de margem máxima separando um conjunto de indivíduos



Fonte: Adaptado de TONG e KOLLER (2002)

Dispondo dos indivíduos do conjunto de treinamento, a busca pelo hiperplano de margem máxima pode ser modelada como um problema de otimização. São utilizados os pontos das margens (os mais difíceis de classificar, chamados de vetores de suporte) para encontrar a equação do hiperplano. Quando o conjunto de dados não é linearmente separável, é possível trabalhar com o mesmo conjunto de dados com um número maior de dimensões para torná-lo separável. Para isto, normalmente são utilizados *Kernels*. A escolha de funções de *Kernel* e verificação se uma delas é válida vai além do escopo deste estudo, o qual pretende apenas introduzir o funcionamento geral dos principais classificadores.

A utilização de SVM tornou-se bastante popular devido à sua robustez mesmo lidando com muitas dimensões (BOIY; MOENS, 2009). Tratando-se de análise de sentimentos, em que comumente há um número grande de *features*, tal característica é imprescindível. Assim, a abordagem normalmente obtém melhores resultados que outros classificadores convencionais como o Bayesiano Ingênuo ou de Entropia Máxima. Por outro lado, seu treinamento é consideravelmente mais lento.

3.4 Implementação e Testes

A fim de testar as técnicas de preparação de documentos e os classificadores descritos anteriormente, foi feita uma implementação que ilustrasse a aplicação de classificadores de análise de sentimentos em conteúdo social, como se pretende utilizar na arquitetura proposta. Em seguida, foram comparadas as taxas de acerto para diferentes configurações.

Para realização dos testes, se faz necessário dispor de conteúdo social previamente classificado com relação à sua polaridade. Foi utilizada a base de dados pública Twitter Sentiment Corpus[27], disponibilizada pela Sander Analytics[28], que contém 5513 *tweets* em inglês classificados manualmente em positivos, neutros ou negativos. Um subconjunto da base (2500 *tweets*) foi randomicamente selecionado e particionado em conjuntos de treinamento (75%) e testes (25%). Devido à natureza dos classificadores estudados, não foi necessário utilizar conjunto de validação.

3.4.1 Preparação Realizada

Antes da extração de *features* e treinamento dos classificadores testados, os *tweets* da base foram pré-processados com o intuito de otimizar o treinamento e a classificação. As etapas da preparação realizada são listadas abaixo, enquanto a Figura 8 mostra o código-fonte da implementação, em Python 2.7[29].

1. Filtro de *stopwords*: a partir de uma lista de *stopwords* da língua inglesa publicamente disponibilizada[30], foram removidas dos *tweets* ocorrências de qualquer uma das palavras contidas na lista;
2. Letras repetidas: repetições consecutivas contendo 3 ou mais letras foram substituídas por apenas 2 letras (por exemplo, “looove” por “loove”). Múltiplos espaços também foram removidos;
3. Remoção de pontuação: vírgulas, pontos finais, interrogações e exclamações foram removidos;
4. Remoção de *hashtags,* usuarios e URL: *hashtags* (‘#’) foram removidos (mas seu conteúdo preservado, por exemplo, “#happy” substituído por “happy”), enquanto URL e menções a usuários (“@usuario”) foram totalmente removidas;
5. *Stemming*: foi utilizado o algoritmo de Porter (PORTER, 1980) para realização de *stemming* . Para investigar a eficácia, os classificadores foram testados com e sem a operação; e
6. Extração das *features*: após as etapas acimas, as palavras restantes são utilizadas individualmente como *features* (testes foram realizados apenas com *unigrams*).

---

27 http://www.sananalytics.com/lab/twitter-sentiment/

28 Empresa norte-americana de análise de dados

29 https://www.python.org/

30 Lista de *stopwords* (*stoplist*) coletadas pela professora Flavia Barros (CIn-UFPE) e publicamente disponibilizada em <http://www.cin.ufpe.br/~if796/aulas/stoplist-ingles>

Figura 8: Implementação da preparação realizada

```
import re
from nltk.stem import PorterStemmer

def preprocessarTweet(tweet):
    tweet = re.sub('((www\.[^\s]+)|(https?://[^\s]+))','*URL*',tweet) # Substituir URL por "*URL*"
    tweet = re.sub('@[^\s]+','*USER*',tweet)                          # Substituir mencao a usuario por "*USER*"
    tweet = re.sub('[\s]+', ' ', tweet)                               # Remocao de multiplos espacos
    tweet = re.sub(r'#([^\s]+)', r'\1', tweet)                        # Eliminacao de caracteres '#'
    return tweet

stopWords = []
def lerStopwords(arquivo):
    stopWords = ['*URL*', '*USER*']     # Tratar *URL* e *USER* como stopwords
    fp = open(arquivo, 'r')
    linha = fp.readline()
    # Incluir, alem de *USER* e *URL*, stopwords lidas do arquivo
    while linha:
        palavra = linha.strip()
        stopWords.append(palavra)
        linha = fp.readline()
    fp.close()
    return stopWords
stopWords = lerStopwords('stopwords.txt')

# Substituir 3 ou mais ocorrencias consecutivas de um caractere por apenas 2
def substituirLetrasRepetidas(s):
    padrao = re.compile(r"(.)\1{1,}", re.DOTALL)
    return padrao.sub(r"\1\1", s)

def extrairFeatures(tweet, aplicarStemming):
    features = []
    palavras = tweet.split()
    for palavra in palavras:
        palavra = substituirLetrasRepetidas(palavra)         # Remove >2 letras iguais repetidas
        palavra = palavra.translate(None, '\'"?,.!&-') # Remove caracteres de pontuacao, hifens, etc
        stemmer = PorterStemmer()
        # Inserir apenas se nao for stopword, ha mais de 1 caractere e inicia com letra do alfabeto
        if(palavra not in stopWords and len(palavra) > 0 and palavra[0].isalpha()):
            if (aplicarStemming):
                features.append(stemmer.stem(palavra.lower()))
            else:
                features.append(palavra.lower())
    return features
```

Fonte: O Autor

3.4.2 Implementação dos Classificadores

Os algoritmos de aprendizagem também foram implementados em Python 2.7. Para os classificadores probabilísticos (Bayes Ingênuo e Entropia Máxima), foi utilizada a biblioteca NTLK (*Natural Language Toolkit*)[31], que possui métodos para treinamento e classificação para tais modelos. Já para a Máquina de Vetores de Suporte, foi utilizada a biblioteca LIBSVM[32], escrita em C++ e com *wrapper*[33] em Python. O código-fonte contendo a implementação dos classificadores é mostrado no Apêndice A.

3.4.3 Resultados

Os resultados dos testes foram resumidos na Tabela 1. Como esperado, a Máquina de Vetores de Suporte alcançou a melhor taxa de acerto, atingindo 70.12% na classificação de

[31] http://www.nltk.org/

[32] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[33] Interface que encapsula e esconde a complexidade da implementação original.

tweets quando não se utilizou *stemming*. Apesar de sua simplicidade, o Bayesiano Ingênuo teve resultados significativamente melhores que o de Entropia Máxima. Esse comportamento está de acordo com os resultados de PANG; LEE; VAITHYANATHAN (2002).

Conforme POTTS (2011) observou, a utilização de *stemming* em mineração de opinião nem sempre traz ganhos na acurácia dos classificadores e sua utilização precisa ser investigada em domínios específicos. Enquanto o Bayesiano Ingênuo teve desempenho melhorado com a utilização de *stemming,* o mesmo não foi verdade para os demais classificadores.

Tabela 1: Taxas de acerto dos classificadores nos testes realizados

Configuração	Bayesiano Ingênuo	Entropia Máxima	Máquina de Vetores de Suporte
Sem *stemming*	63.93%	61.39%	70.12%
Com *stemming*	68.17%	58.70%	68.42%

Fonte: O Autor

De forma geral, fica claro que a utilização de aprendizagem de máquina está longe de obter resultados ótimos, devido à natureza subjetiva do problema de análise de sentimentos. Contudo, os resultados ainda assim são bastante satisfatórios, ultrapassando 70% de acerto lidando com três classes. É importante ressaltar que foi utilizado um conjunto de dados relativamente pequeno e com *tweets* de diversos domínios. Tratando-se de um domínio específico (como uma marca ou corporação) e uma base de dados maior, resultados ainda melhores poderiam ser atingidos.

3.5 Considerações Finais

Apesar de altamente complexo, o problema de classificar a polaridade de fragmentos de texto tem atraído bastante atenção e, cada vez mais, surgem na literatura abordagens capazes de alcançar altas taxas de acerto na classificação de sentimentos, conforme ilustrado nos testes realizados. Assim, é válido dizer que atualmente é possível tirar vantagem deste tipo de técnica para agregar automaticamente informação sobre a opinião expressa em diferentes contextos. Entre eles, o contexto de redes sociais online, de onde é possível extrair dados com relativa facilidade, como mostrado no capítulo anterior.

Dispondo dos dados brutos e de estratégias para enriquecê-los por meio de análise de sentimentos, se fazem necessárias formas de armazenar, visualizar e consultar eficientemente este tipo de informação. Com este intuito, ferramentas de Business Intelligence (BI), já populares para tratamento de largas quantidades de dados não-estruturados, podem ser extremamente úteis. O capítulo seguinte explora como BI pode ser usada para inúmeras finalidades, como integração de dados de diferentes plataformas, observação de tendências e predições e consultas complexas que seriam extremamente difíceis com utilização de bancos de dados convencionais.

CAPÍTULO 4

Business Intelligence

Business Intelligence (BI) e a grande área de análise de *big data* têm ganhado cada vez mais importância, tanto na indústria como na comunidade acadêmica. As oportunidades associadas com a análise inteligente de dados em diferentes contextos atraiu o interesse de organizações que buscam ferramentas capazes de otimizar tanto a análise de suas informações como de dados externos e heterogêneos, que permitam entender o mercado e guiar suas decisões corporativas.

Este capítulo pretende fazer um estudo sobre a área de Business Intelligence. Inicialmente, é feita uma introdução sobre o tema. Em seguida, o texto trata de *Data Warehousing*, abordagem para materialização de dados diretamente relacionada ao conceito de BI. Por fim, é feita uma análise do porquê de haver um grande potencial para combinação de ferramentas de BI com o conteúdo de redes sociais online.

4.1 Introdução

Um dos bens mais importantes de qualquer organização é a sua informação (KIMBALL; ROSS, 2011). Não apenas a gerência de suas informações internas, mas a forma como uma organização adquire informações externas para dar suporte à tomada de decisões é um fator indispensável em um ambiente fortemente competitivo. Atualmente, o acesso a dados dispersos e a capacidade de integrar e processar informações são essenciais para desenvolver atividades comerciais. Estas informações devem ser relevantes e coerentes para que permitam um bom planejamento estratégico e, para tal, é necessário que sejam aproveitadas as oportunidades que surgem a partir da utilização correta de tecnologias disponíveis.

Neste contexto, uma área que tem ganhado destaque na extração, integração e visualização de dados é a de *Business Intelligence* (BI). Este termo foi utilizado pela primeira vez em 1958, definido como um "conjunto de atividades para diversos propósitos, providas por um sistema inteligente capaz de assimilar relações entre fatos com o intuito de guiar ações para atingir um objetivo desejado" (LUHN, 1958). Posteriormente, o termo passou a ser utilizado de forma mais ampla, incluindo quaisquer soluções ou ferramentas capazes de transformar dados brutos em informações úteis para alguma organização.

Contudo, foi só na década de 1990 que o termo ganhou popularidade (POWER, 2008). A crescente busca por sistemas de suporte à decisão e capazes de lidar com grandes quantidades de dados popularizou a área de BI e fez com que a atividade de pesquisa na área aumentasse consideravelmente. Gradualmente, BI passou de um conjunto de técnicas ingênuas para uma abordagem bem fundamentada de extração e processamento de informação (GOLFARELLI et al., 2004).

Atualmente, o conceito de BI envolve duas atividades primárias: aquisição de dados (*"data in"*) e visualização de dados (*"data out"*) (WATSON; WIXON, 2007). O processo de aquisição de dados envolve a extração, transformação e carga de dados em um repositório central. Esses dados, que podem ser extraídos tanto de fontes externas e heterogêneas (redes sociais online ou documentos na Web, por exemplo) como de fontes internas da própria organização (como diferentes bancos de dados de uma mesma empresa), e são normalmente carregados em um repositório orientado a tempo e assunto, chamado de *data warehouse*.

A segunda preocupação de BI é com o acesso aos dados previamente armazenados no *data warehouse,* bem como a realização de processamentos e análises sobre este conteúdo. Construção de relatórios, consultas e análises multidimensionais, mineração de dados, realização de predições e observação de tendências são exemplos de conceitos explorados em BI para permitir uma visualização inteligente da informação. Atualmente, há uma diversidade imensa de ferramentas para visualização e análise de informação de *data warehouses* (MUNDY; THORNTHWAITE, 2011).

Ferramentas de visualização de BI normalmente são construídas de forma que possam ser utilizadas por usuários de negócios e, portanto, devem ser naturalmente amigáveis e intuitivas. Essa simplicidade, aliada à natureza integradora de *data warehouses* e a necessidade de corporações de entender e analisar dados não-estruturados de redes sociais online são fortes motivações para utilização de BI no contexto social. Além deste contexto específico, a área também tem se popularizado na indústria como um todo: de acordo com SALLAM et al. (2015), o mercado de BI apresentou um dos maiores crescimentos dentre todo o mercado de software nos últimos anos. Assim, é claro o interesse neste tipo de solução e fornecedores de software têm investido bastante no desenvolvimento de tecnologias competitivas e cada vez mais completas.

4.2 *Data Warehousing*

O processo de aquisição de dados em BI é chamado por alguns autores de *Data Warehousing* devido à forte conexão entre BI e a utilização de *Data Warehouses* (DW). Os conceitos estão tão fortemente relacionados que alguns autores combinam os termos em uma única sigla: BI/DW ou BIDW (GOLDEN, 2013). DW são definidos como "Uma coleção de dados orientada a assunto, integrada, variante no tempo e não-volátil, usada para suporte do processo de tomada de decisões de uma organização" (INMON, 2005). Estas características são exploradas com maior profundidade na lista abaixo:

1. Orientados a assunto - DW foram pensados para viabilizar análises de dados para contextos específicos;
2. Integrados - DW devem ser capazes de integrar dados de diferentes fontes em um formato consistente, solucionando problemas de integração de dados como conflito de nomes de entidades e inconsistências em unidades de medida utilizadas;

3. Não-voláteis - Uma vez que o dado é inserido no *data warehouse*, não deve ser alterado ou removido;
4. Variantes no tempo - Para permitir análise de tendências e predições, DW guardam dados históricos e como o dado variou ao longo do tempo.

4.2.1 Extração, Transformação e Carga

Sistemas de integração de dados, como DW, são normalmente classificados com relação à abordagem tomada para interagir com os dados da fonte, podendo ser virtual ou materializada. Na primeira, as informações são extraídas diretamente das fontes, quando necessárias, sem haver cópia dos dados. Na segunda, as informações são recuperadas, integradas e armazenadas em um repositório central, evitando que seja necessário acessar as fontes cada vez que for preciso realizar uma consulta (SALGADO; LÓSCIO, 2001).

A partir da observação da definição de DW e de suas características, fica claro que este é um exemplo de abordagem materializada, já que os dados das fontes são armazenados em um repositório central. Assim, os dados analisados em um projeto de BI são sempre recuperados do DW e não das fontes diretamente. Para garantir consistência e disponibilidade, é necessário carregar os dados das fontes regularmente. O processo de extração das fontes, processamento dos dados e carga no DW é comumente chamado de ETL, sigla para Extração, Transformação e Carga (do inglês, *Extract, Transform and Load*).

ETL deve ser visto como um processo completo, não restrito ao contexto de *Data Warehousing*, e não como um conjunto de três processos independentes (ORACLE, 2002). Em um contexto de integração de dados em DW, o processo de ETL deve ter as etapas bem definidas e fortemente integradas. No processo de extração, é definido de onde e como os dados serão recuperados (o tipo de fonte de dados pode variar bastante). Assim, os dados das bases (ou parte deles) podem ser transferidos para o ambiente do DW, a partir onde o sistema pode operar de forma independente das fontes.

No processo de transformação dos dados, são realizados possíveis ajustes nos dados transferidos, de forma a adequá-los à modelagem do DW. Exemplos de operações de ajuste possíveis são tradução, conversão de unidades, junção de dados, entre outros. Além destes ajustes, também é possível agregar conhecimento aos dados coletados. Por exemplo, em um contexto social, o processo ETL poderia contar com um módulo classificador que acrescentaria informação sobre a polaridade de publicações, conforme visto no Capítulo 3. Um estudo de caso mostrando a utilização de análise de sentimentos em um processo ETL será mostrado no Capítulo 5 deste trabalho.

Por fim, os dados são carregados no DW na fase de carga. A forma como a carga ocorre varia de acordo com as necessidades do sistema, podendo ser feita em processos *batch*, por linha, em tempo real, entre outros (ORACLE, 2002). Além disso, é definida a temporização com a qual a carga deve acontecer. Fornecedores normalmente incluem ferramentas de ETL em suas soluções de BI para facilitar o processo na construção e manutenção de DW.

4.2.2 OLAP vs. OLTP

Diferentes dos bancos de dados transacionais, altamente normalizados a fim de facilitar operações de inserção, atualização e remoção, *data warehouses* têm o intuito de dar suporte a consultas complexas, mas sem se preocupar com a atualização dos dados já inseridos. As formas como os dados são processados em sistemas transacionais e sistemas analíticos (como DW) são chamadas, respectivamente, de OLTP (*OnLine Transaction Processing*) e OLAP (*OnLine Analytical Processing*). As principais diferenças entre estes dois tipos são resumidas no Quadro 2.

Quadro 2: Comparação de características dos processamentos OLTP e OLAP

OLTP	OLAP
Transações numerosas e curtas	Análise gerencial
Dados detalhados e distribuídos	Dados consolidados
Consultas pré-definidas; pouco volume de dados	Consultas *ad hoc*; alto volume de dados
Alta concorrência	Baixa concorrência
Mais normalização	Mais redundância
Banco de dados reflete o estado atual	Reflete o estado da última carga
Foco: armazenamento, confiabilidade, desempenho e disponibilidade	Foco: recuperação, visualização, flexibilidade de consultas

Fonte: (SALGADO et al., 2015)

4.2.3 Modelagem de *Data Warehouses*

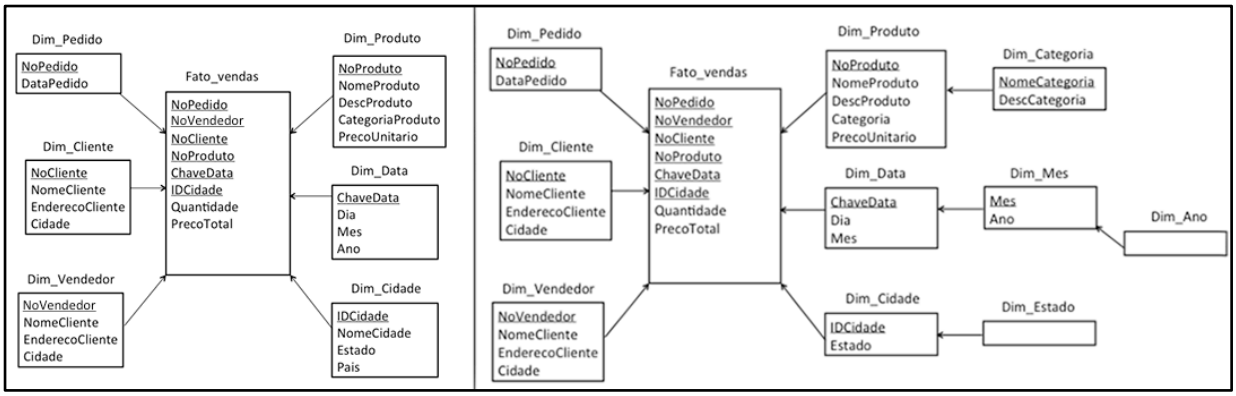
Esquemas de aplicações OLAP também são diferentes de esquemas tradicionais OLTP. Diagramas entidade-relacionamento e técnicas de normalização, populares em ambientes OLTP, são normalmente inapropriados para sistemas de suporte à decisão em BI, nos quais normalmente se carregam e consultam largas quantidades de dados (CHAUDHURI; DAYAL, 1997). Assim, modelagens de DW são normalmente diferentes de bancos de dados tradicionais e requerem cuidados especiais.

Na modelagem de DW, utilizam-se dois tipos de tabelas: tabelas-fato e dimensão. Nas tabelas-fato, utiliza-se um ponteiro (chave estrangeira) para cada uma das tabelas-dimensão relacionadas e possivelmente métricas numéricas sobre algum assunto da organização, tratado na tabela, como valores de vendas, por exemplo. Já as tabelas-dimensão contêm informações que descrevem os registros da tabela-fato. A Figura 9 mostra dois exemplos de modelagens de DW contendo tabelas fato e dimensão e como estas se relacionam.

Há dois possíveis tipos de esquemas que se diferenciam de acordo com a forma com que as tabelas-fato e dimensão se relacionam. No primeiro, chamado de estrela, há uma única tabela-fato que relaciona-se com uma ou mais tabelas-dimensão. Neste tipo de esquema, não há suporte explícito para hierarquia de dimensões. Isto é, não é possível

relacionamento entre duas tabelas-dimensão. O segundo, chamado de floco-de-neve, funciona como um refinamento do modelo estrela, em que as dimensões são normalizadas em uma estrutura hierárquica. A normalização das tabelas-dimensão traz ganhos na manutenção das dimensões, pois reduz o tamanho das tabelas e as redundâncias. Contudo, a estrutura não-normalizada do modelo estrela muitas vezes torna-se mais apropriada para consultas complexas, o que torna este modelo mais popular em relação ao outro (CHAUDHURI; DAYAL, 1997). A Figura 9 compara um mesmo esquema no modelo estrela e no floco-de-neve.

Figura 9: Exemplos de duas modelagens de DW: Estrela e Floco-de-Neve



Fonte: Adaptado de CHAUDHURI e DAYAL (1997)

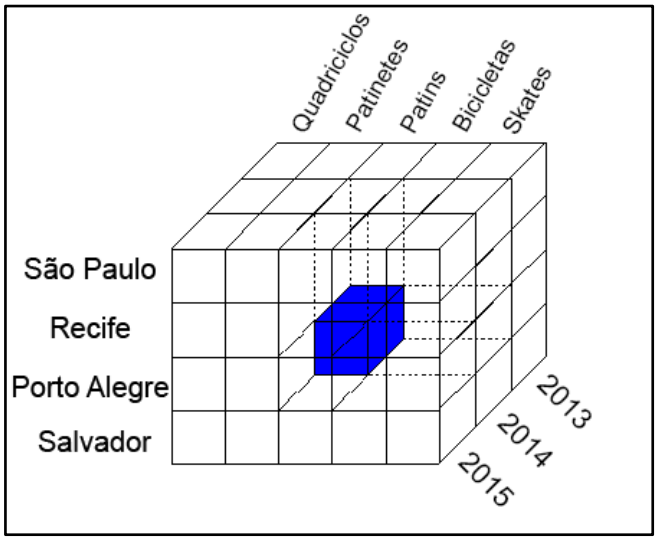
É importante ressaltar que apesar de sistemas OLTP normalmente serem modelados de forma a minimizar redundâncias, isto não necessariamente se aplica à modelagem de *data warehouses*. Estes normalmente são deliberadamente modelados de forma que redundâncias estão intrínsecas à arquitetura: parte dos dados podem ser vistos a partir de outros dados e, logo, redundantes do ponto de vista lógico. Contudo, esta redundância é vista como algo positivo, visto que acaba por aumentar a performance de consultas complexas (GUTIÉRREZ; MAROTTA, 2001). Além disso, como tipicamente não há atualização dos dados, a existência de redundâncias não traz preocupações associadas à necessidade de atualizar um mesmo dado disperso em várias localizações.

4.2.4 Cubos OLAP

Um conceito fundamental tratando-se de BI e OLAP é a representação multidimensional dos dados em cubos. Nesta representação, os dados são dispostos como em um cubo, no qual as células contêm valores registrados em uma tabela-fato e os lados definem algumas dimensões com as quais o registro se relaciona. Por exemplo, para o caso da modelagem da Figura 9, seria possível construir um cubo com as dimensões Produto, Cidade e Data e cada célula representaria informações (quantidade e preço total) para cada possível combinação dessas dimensões. A Figura 10 mostra um exemplo de cubo OLAP construído

para este contexto de vendas. Neste caso, a célula destacada contém métricas sobre as vendas de bicicletas em Recife no ano de 2014.

Figura 10: Exemplos de cubo OLAP



Fonte: Adaptado de ORACLE (2002)

Representar dados dessa maneira é bastante útil, porque permite que sejam realizadas inúmeras combinações de cubos. Assim, cada cubo representa uma diferente visão sobre os dados e podem responder consultas com muita velocidade (KIMBALL; ROSS, 2011). Para o caso do cubo mencionado anteriormente, poderia ser consultado, por exemplo, o total de vendas para determinados produto e cidade em um certo período de tempo e o sistema poderia responder rapidamente, por já ter essa informação agregada no cubo construído. Este tipo de operação, em um ambiente OLTP e sob uma modelagem altamente normalizada, poderia ser extremamente custoso para o sistema.

Apesar de os tipos de análise disponíveis variarem de acordo com o fornecedor escolhido para o projeto de BI, alguns são normalmente explorados por todos. Algumas operações convencionais sobre cubos OLAP são definidas por INMON (2005) e são comumente exploradas em ferramentas de análise multidimensional. São elas:

1. *Drill-down* - Aumento do nível de detalhamento da informação (e diminuição da granularidade). Pode ser feita descendo no nível de hierarquia de dimensões (por exemplo, transformando uma dimensão “ano” em “mês”) ou acrescentando uma nova dimensão;
2. *Roll-up* - Operação reversa do *drill-down*, diminuindo o nível de detalhamento;
3. *Slice* - Seleção de um subconjunto do cubo a partir da atribuição de um valor específico para uma das dimensões. O resultado da operação é a criação de um novo cubo com uma dimensão a menos que o original;
4. *Dice* - Criação de um subcubo a partir da seleção de um subconjunto das dimensões de um cubo original.

4.2.5 Visualização de Informação

A última preocupação típica de um sistema de BI diz respeito a ferramentas de acesso aos dados armazenados. A complexidade das funcionalidades a serem utilizadas variam fortemente de acordo com a natureza dos dados e das análises a serem feitas. Há, disponíveis no mercado, tecnologias cobrindo tanto requisitos simples como consultas *ad hoc* ou outros mais complexos como ferramentas de mineração de dados e predição. Estes sistemas finais de consultas e análise de informações que se fundamentam em uma arquitetura de *Data Warehouses* são comumente chamados de sistemas de *reporting* (MUNDY; THORNTHWAITE, 2011).

Depois de construção do DW, o sistema ETL que o alimenta e uma infraestrutura de informação dimensional sólida, a utilização ou criação de sistemas de *reporting* é simples. Atualmente há uma variedade grande de fornecedores de *software* que entregam sistemas para construção de DW e soluções integradas para explorar os dados armazenados. Em SALLAM et al. (2015) é mostrado um relatório anual realizado pela Gartner[34], empresa norte-americana de consultoria na área de tecnologia, contendo informações o mercado e comparando fornecedores do mercado de BI. Segundo a pesquisa, os fornecedores que lideram atualmente no mercado de BI incluem IBM[35], Microsoft[36], Oracle[37], SAS[38] e Tableau Software[39].

Apesar de tipicamente haver uma linguagem de consulta utilizada internamente para consultar cubos OLAP, como a MDX[40], introduzida pela Microsoft, é comum que as ferramentas ofereçam formas simplificadas de interagir com dimensões realizar operações sobre cubos para visualização de informações em um ambiente OLAP. Assim, é possível construir relatórios detalhados sem utilização obrigatória de linguagens de consulta, mas sim selecionando opções em interfaces bastante intuitivas. O relatório da Gartner que analisa as tendências do mercado de BI inclusive destaca a experiência do usuário de negócio como uma das 17 capacidades fundamentais que uma ferramenta de BI deve atender, mostrando que uma das tendências da área é aproximar-se cada vez mais deste tipo de usuário.

4.3 Business Intelligence e Redes Sociais Online

Muitas organizações estão começando a perceber que monitoramento e análise de conteúdo social precisa ser parte de sua estratégia de negócio. Dados sociais podem oferecer informações tão relevantes quanto dados internos da empresa. Contudo, dada a velocidade com a qual a informação é produzida em redes sociais, para uma análise bem sucedida, é

[34] http://www.gartner.com/technology/home.jsp
[35] http://www.ibm.com/
[36] http://www.microsoft.com/
[37] http://www.oracle.com/
[38] http://www.sas.com/
[39] http://www.tableau.com/
[40] *MultiDimensional eXpressions*, linguagem de consulta para bancos de dados OLAP.

necessária uma abordagem que se comporte bem lidando com integração de grandes quantidades de dados heterogêneos.

Com o crescimento desta demanda de analisar conteúdo de redes sociais online, surgiu uma necessidade cada vez maior de correlacionar conteúdo social com informações sobre consumidores e integrar dados de redes sociais com sistemas de informação (ORACLE, 2013). Fornecedores de soluções de BI, ao observar esta necessidade e o interesse por parte de organizações em analisar conteúdo social, passaram a se preocupar em tornar suas ferramentas naturalmente integradas com plataformas sociais, passando a tratá-las como fontes de dados de processos ETL, da mesma forma que tratam-se bancos de dados convencionais.

Ao coletar e analisar dados sociais correlacionados com os dados da empresa, as organizações ganham uma visão melhorada sobre seus consumidores. Este tipo de análise oferece entendimentos que permitem ir além de conclusões simples como "Produto X tem boas vendas na região sul" para complexas, como “Por que o produto X tem boas vendas na região sul?”. Uma aplicação de BI dá uma representação visual deste tipo de dados, facilitando a identificação de tendências e entendimento do mercado. Assim, este tipo de solução pode alavancar o desenvolvimento de produtos e *marketing* de organizações.

Além de facilitar a visualização e análises, outro fator que aproxima bastante BI de redes sociais online é o fato de que tipicamente há uma quantidade imensa de conteúdo social a ser analisado. Ao monitorar e armazenar dados de uma certa marca ou um certo produto em redes sociais, é comum que seja recuperada uma grande quantidade de conteúdo. Como aplicações OLAP têm desempenho muito melhor que bancos de dados transacionais para analisar largas quantidades de dados (INMON, 2005), estas acabam tornando-se alternativas melhores para análise de redes sociais.

Por exemplo, no estudo de caso mostrado no Capítulo 5, em apenas um dia extraindo dados de uma única plataforma para um único termo de consulta, foram recuperadas mais de 3 mil publicações. Ao se pensar em um contexto maior, com mais plataformas sociais, meses de monitoramento e vários termos sendo analisados, este número cresce muito rapidamente. Realizar operações de álgebra relacional de bancos transacionais OLTP para lidar com potencialmente bilhões de registros torna-se impraticável. Assim, a busca por abordagens alternativas, como *Business Intelligence*, acaba sendo inevitável.

4.4 Considerações Finais

Ainda há muito espaço para crescimento do mercado de BI, visto que este está constantemente se adaptando ao cenário atual da tecnologia. A complexidade aumenta com a introdução de novos tipos de fontes de dados (tais como a nuvem, eventos em tempo real, redes sociais online, sensores, entre outros) e novos tipos de análise (como mineração de opinião e outras de técnicas aprendizagem de máquina). Assim, novas oportunidades continuam emergindo para agregar valor para soluções de BI e, em virtude disso, os líderes de iniciativas de BI estão sob uma pressão cada vez maior para identificar tais oportunidades e conquistar os consumidores, que têm interesse cada vez maior neste mercado.

Uma destas oportunidades se mostra com o rápido crescimento da utilização de redes sociais online. A ampla adoção destas plataformas no mundo traz consigo uma imensa quantidade de dados que podem ser transformados em informações extremamente úteis para organizações e, assim, há uma forte tendência de combinação de Business Intelligence e redes sociais. O próximo capítulo mostrará um estudo de caso ilustrando como é possível combinar estes conceitos por meio de uma aplicação capaz de extrair dados sociais, processá-los com técnicas de mineração de opinião estudadas e, por fim, armazená-los num *Data Warehouse*, o qual pode ser utilizado para realizar consultas e relatórios.

CAPÍTULO 5

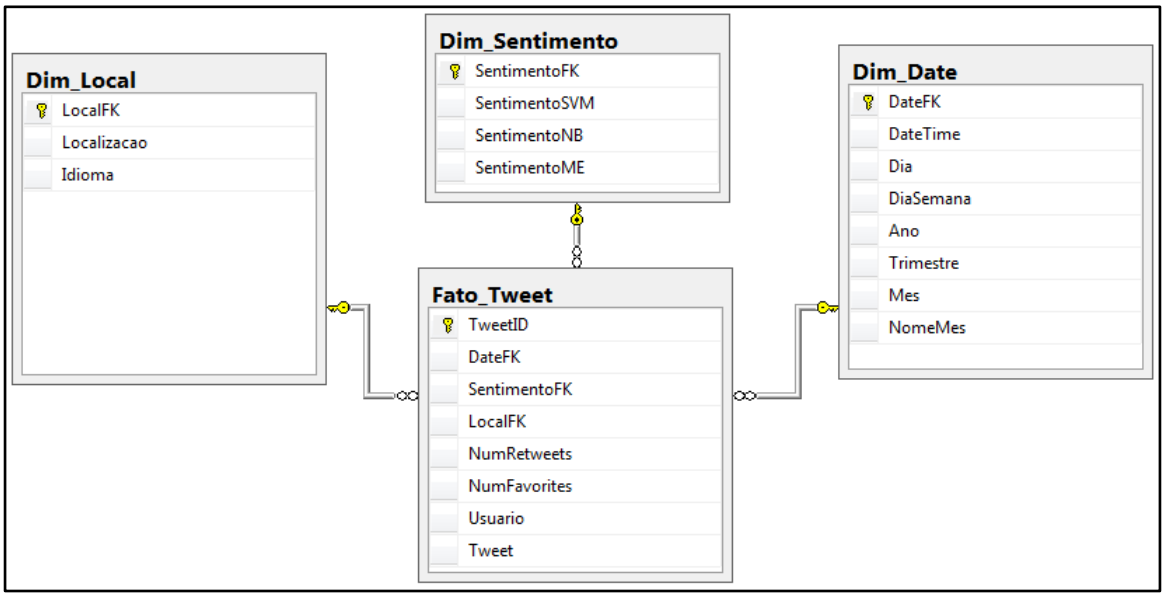
Estudo de Caso

Após um extenso estudo sobre extração de dados de Redes Sociais Online (RSO), análise de sentimentos e Business Intelligence (BI), o trabalho mostra um estudo de caso que ilustra como estes conceitos poderiam ser utilizados em um contexto real. Este capítulo mostrará uma implementação de uma arquitetura baseada em BI e mineração de opinião para construção de um *Data Warehouse* (DW) com *tweets* extraídos do Twitter[41], enriquecidos por meio da agregação de informação sobre suas polaridades. Será mostrado em detalhes como se deu o processo de extração, transformação e carga dos dados no DW. Por fim, são dados alguns exemplos de formas de consultar e analisar a informação armazenada.

5.1 Modelagem do *Data Warehouse*

Antes mesmo de pensar na extração de dados, é preciso que seja criada a modelagem do DW que irá armazenar os dados sociais. Afinal, é preciso conhecer o esquema a ser utilizado para saber exatamente quais dados (dentre os disponibilizados pelo Twitter) devem ser salvos. A Figura 11 mostra o diagrama do esquema construído para armazenar *tweets*, alguns metadados associados e polaridade obtida por cada um dos classificadores estudados. Como o objetivo deste estudo de caso era armazenar e analisar apenas dados do Twitter, a modelagem foi bastante simples em relação a cenários reais de utilização de DW.

Figura 11: Esquema do *Data Warehouse* construído para armazenar *tweets* e dados associados



Fonte: O Autor

[41] http://www.twitter.com/

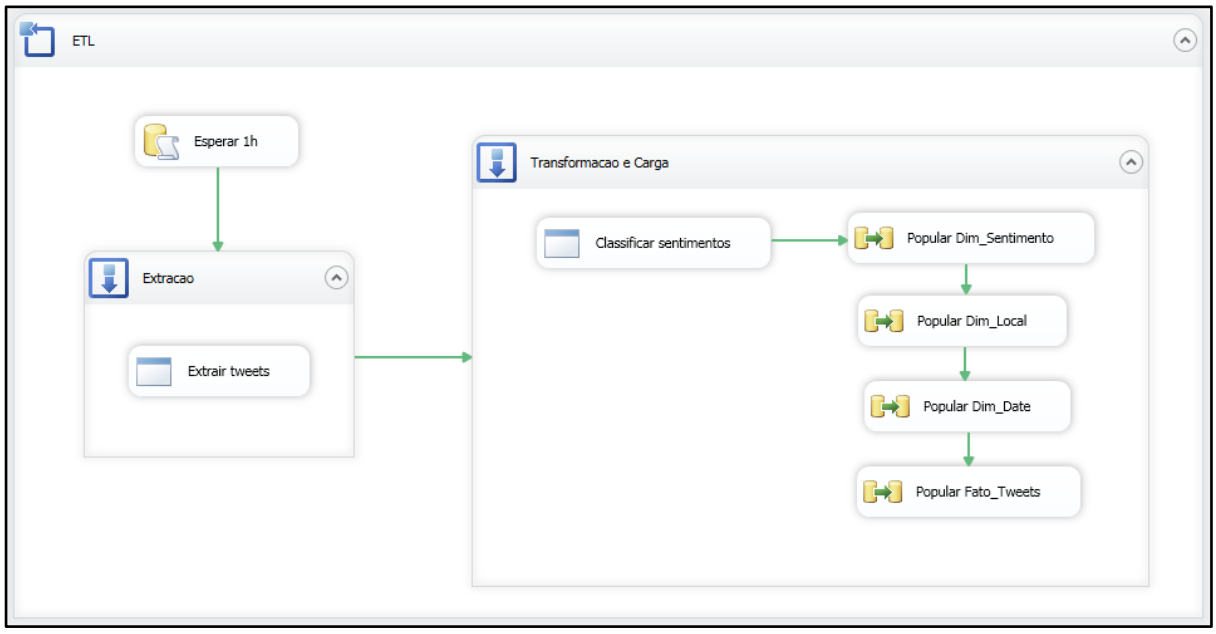
Para modelagem e criação do DW, foi utilizado o Microsoft SQL Server 2012[42]. O SQL Server é um dos SGBD[43] mais utilizados atualmente e permite tanto criação de bancos de dados (módulo *Database Engine*) como análises sobre cubos OLAP (módulo *Analysis Services*). Nesta etapa, foi utilizado apenas o *Database Engine* para criação das tabelas fato e dimensão e dos relacionamentos entre estas, conforme mostrado na figura anterior.

5.2 Extração, Transformação e Carga

Para coordenar o processo ETL, foi utilizado um componente do SQL Server que fornece apoio à construção de processos de migração e transformação de dados, o SQL Server Integration Services (SSIS). A ferramenta permite que seja construído um fluxo de operações que incluem leitura de dados de alguma fonte (banco de dados, arquivo de texto, página Web, entre outros), tratamento sobre estes dados (gerar colunas derivadas, agregar conhecimento a partir de operações externas, filtragem dos dados, entre outros) e carga de dados (para arquivos externos ou bancos de dados, inclusive *data warehouses*).

O esquema geral do processo criado é mostrado na Figura 12. Todo o fluxo está contido numa caixa de repetição ("*For container*"), que repete seu conteúdo enquanto uma dada condição for verdadeira. Neste caso, foi utilizada a condição TRUE para que o processo fosse repetido indefinidamente, até que explicitamente interrompido. Em um contexto real, um servidor poderia ser alocado para executar um processo semelhante e não precisaria ser monitorado.

Figura 12: Esquema do processo ETL criado no SQL Server Integration Services



Fonte: O Autor

[42] http://www.microsoft.com/en-us/server-cloud/products/sql-server/features.aspx

[43] Sigla para Sistema de Gerenciamento de Banco de Dados.

O processo inicia com um componente *Execute SQL Task*, que roda um script Transact-SQL[44] para pausar o processo por uma hora (este tempo pode ser facilmente ajustado para se adequar a diferentes necessidades). Para causar o atraso, foi utilizada a função WAITFOR DELAY, nativa do Transact-SQL, que faz com que o processo seja pausado por um dado intervalo de tempo.

5.2.1 Extração de Dados do Twitter

Para extrair dados do Twitter, uma de suas API (REST API[45]) foi utilizada (detalhada na subseção 2.3.1 deste trabalho). A linguagem de programação utilizada na extração foi Python 2.7[46] e a implementação final bastante semelhante com a mostrada anteriormente, acrescentando apenas a extração de outros metadados disponíveis no arquivo JSON retornado na consulta à API.

O termo utilizado para buscar *tweets* foi "Surface book", o mais recente computador lançado pela Microsoft[47]. Os dados recuperados (100 *tweets* mais recentes e dados associados, para cada execução) são salvos num arquivo de texto CSV[48] para que possam ser devidamente transformados antes de carregados no DW. Um trecho de um dos arquivos criados no final do processo de extração é mostrado na Figura 13.

Figura 13: Trecho de arquivo criado no processo de extração de dados do Twitter

```
1  tweetID | usuario | localizacao | idioma | data | tweet | retweets | favorites
2  664152439604568064 | casuallynorm | North Carolina, USA | en | Tue Nov 10 18:47:35 +0000 2015 | @TheNewLou @zcichy Until the Mac
   goes Surface Book form factor, with Mac OS when attached and iOS when detached. Ba-Boom! | 0 | 0
3  664151655374389249 | _AUSSON | Chicago, IL | en | Tue Nov 10 18:44:28 +0000 2015 | Critics Love Microsoft's Surface Book https://
   t.co/oRvVq3shob | 0 | 0
4  664151428093624320 | Karlitoz007 | Lima, Perú | en | Tue Nov 10 18:43:34 +0000 2015 | Painting with pixels: The camera
   technology of the Surface Pro 4 and Surface Book. #Windows https://t.co/WPOwqoEcAi | 0 | 0
5  664143474699059200 | sekuchceiwau198 |  | en | Tue Nov 10 18:11:58 +0000 2015 | RT @EnriqueSandov: Surface pro book, pretty
   impressive hardware | 1 | 0
6  664141367355047936 | SurfaceBook_ | Munich, Germany | en | Tue Nov 10 18:03:35 +0000 2015 | Fallout 4 on a Surface Book.... so
   far, so good. https://t.co/si7aThGYUe #surfaceBook #microsoftSurfaceBook | 0 | 0
```

Fonte: O Autor

No fluxo do processo ETL, o processo de extração é realizado por meio da utilização do componente *Execute Process*, que permite a execução de uma aplicação externa. O componente carrega um executável construído a partir da aplicação Python implementada para extrair dados do Twitter, descrita na seção 5.2. Apesar de o *Integration Services* não ser diretamente compatível com aplicações Python, foi possível transformar a aplicação em um

[44] Extensão proprietária da Microsoft para a linguagem SQL, utilizada no SQL Server.

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

[46] https://www.python.org/

[47] http://www.microsoft.com/

[48] Sigla para *Comma-Separated Value*, extensão de arquivo que armazena dados tabulares em arquivos de texto através da utilização de caracteres delimitadores (normalmente vírgulas) para separar valores de diferentes colunas.

executável, utilizando a biblioteca Py2Exe[49] e utilizar o executável gerado. Quando a execução é concluída, o arquivo de texto contendo os últimos *tweets* recuperados é atualizado e o processo segue.

5.2.2 Análise de Sentimentos

O módulo de transformação e carga inicia com a execução de outra aplicação externa. Neste caso, são executados os classificadores (cuja implementação é mostrada no Apêndice A). Novamente, foi criado um executável para a aplicação previamente descrita. Esta aplicação lê o arquivo CSV contendo os *tweets* salvos no componente anterior e acrescenta 3 colunas, referentes às classificações de cada um dos algoritmos estudados (Bayesiano Ingênuo, Entropia Máxima e Máquina de Vetores de Suporte). Por fim, o arquivo é salvo e são iniciados os fluxos de dados de transformação e carga específicos para cada tabela. A Figura 14 mostra o mesmo trecho de *tweets* da Figura 13, após a execução dos classificadores (classificações destacadas).

Figura 14: Trecho de arquivo anterior, após etapa de análise de sentimentos

```
1  tweetID | usuario | localizacao | idioma | data | tweet | retweets | favorites | sentimentoME | sentimentoNB | sentimentoSVM
2  664152439604568064 | casuallynorm | North Carolina, USA | en | Tue Nov 10 18:47:35 +0000 2015 | @TheNewLou @zcichy Until the Mac
   goes Surface Book form factor, with Mac OS when attached and iOS when detached. Ba-Boom! | 0 | 0 | neutral | negative | neutral
3  664151655374389249 | _AUSSON | Chicago, IL | en | Tue Nov 10 18:44:28 +0000 2015 | Critics Love Microsoft's Surface Book https://
   t.co/oRvVq3shob | 0 | 0 | positive | positive | positive
4  664151428093624320 | Karlitoz007 | Lima, Peru | en | Tue Nov 10 18:43:34 +0000 2015 | Painting with pixels: The camera
   technology of the Surface Pro 4 and Surface Book. #Windows https://t.co/WPOwqoEcAi | 0 | 0 | positive | positive | neutral
5  664143474699059200 | sekuchceiwau198 | New York, NY | en | Tue Nov 10 18:11:58 +0000 2015 | RT @EnriqueSandov: Surface pro book,
   pretty impressive hardware | 1 | 0 | positive | positive | positive
6  664141367355047936 | SurfaceBook_ | Munich, Germany | en | Tue Nov 10 18:03:35 +0000 2015 | Fallout 4 on a Surface Book.... so
   far, so good. https://t.co/si7aThGYUe #surfaceBook #microsoftSurfaceBook | 0 | 0 | positive | positive | positive
```

Fonte: O Autor

5.2.3 Transformações e Cargas no *Data Warehouse*

O módulo de transformação e carga segue com os tratamentos de dados e cargas específicas para cada tabela do DW. Cada um dos processos individuais é representado no processo ETL por um componente *Data Flow Task*, que permite o encapsulamento de um fluxo de migração e tratamento de dados.

De forma geral, os fluxos de transformação e carga para todas as tabelas são semelhantes, havendo apenas algumas particularidades específicas para algumas. O fluxo geral realizado para as tabelas é listado abaixo, e o refinamento de cada um dos *Data Flow Tasks* usados é mostrado em seguida, na Figura 15.

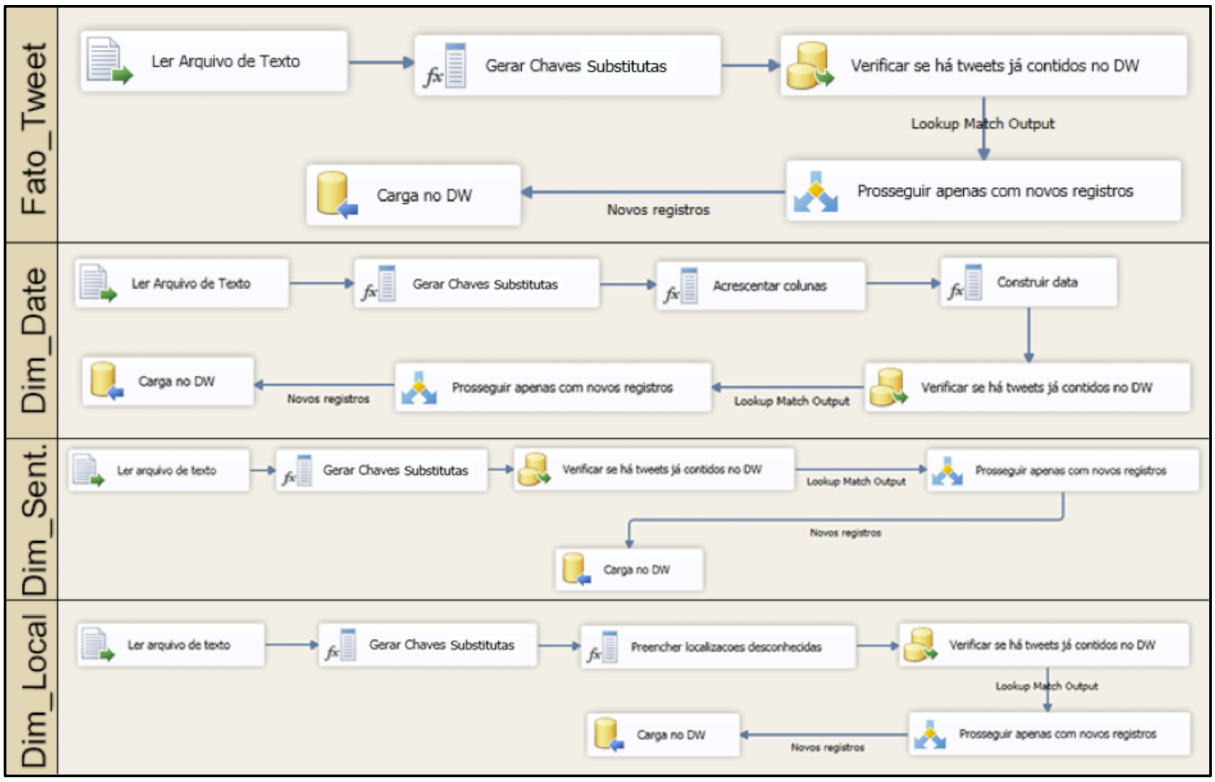
1. Ler arquivo de texto: O arquivo gerado após análise de sentimentos dos tweets é lido. Para isso, é usado um componente de leitura de arquivo (*Flat file source*);
2. Gerar chaves substitutas (*Surrogate Keys*): O componente *Derived Column*, usado para criação de colunas derivadas a partir de dados lidos do arquivo ou de outras

[49] http://py2exe.org/

colunas, é usado para gerar as chaves substitutas, utilizadas como chaves estrangeiras da tabela-fato. Na prática, o valor destas chaves não importa, já que sua única função é identificar unicamente linhas das dimensões e relacioná-las com registros da tabela-fato (INMON, 2005). Assim, para geração das chaves o campo tweetID, que identifica unicamente tweets, é replicado e usado como valor das chaves. É importante lembrar que apesar de essa replicação causar redundância (várias colunas contendo o mesmo valor), em modelagens de DW normalização e prevenção de redundâncias não são preocupações e é comum haver redundância proposital;

3. Verificar se há *tweets* já contidos no DW: Nesta etapa, utiliza-se o componente *Lookup* para realizar uma checagem que garante que os *tweets* contidos no arquivo lido não estão inseridos no DW, para evitar registros repetidos. Essa verificação é necessária, pois é possível que haja interseção entre os 100 *tweets* mais recentes da extração atual e os *tweets* da extração anterior;
4. Prosseguir apenas com novos registros: Após a verificação anterior, é utilizado um componente *Conditional Split*, que permite a criação de fluxos condicionais, para garantir que apenas os novos registros (obtidos na etapa anterior) prosseguirão para a carga;
5. Carga no DW: Finalmente, os registros são carregados no DW. A carga é feita utilizando o componente *OLE DB Destination*, que carrega conteúdo num banco do SQL Server.

Figura 15: Tratamentos de dados (transformações) e cargas no *Data Warehouse* individuais de cada tabela



Fonte: O Autor

Observando a Figura 15, é possível ver que os fluxos da tabela-fato e da dimensão Sentimento são idênticos ao descrito anteriormente. Para as demais tabelas, alguns tratamentos adicionais foram necessários. Na dimensão Local, foi utilizado um componente adicional do tipo *Derived Column*, "Preencher localizações desconhecidas". Seu objetivo é padronizar registros em que a localização não foi fornecida pelo usuário, substituindo o valor NULL ou *strings* vazias com o valor "Desconhecido".

Na dimensão Date, foram utilizados dois componentes adicionais do tipo *Derived Column*. O primeiro, Acrescentar colunas, transforma a única coluna data do arquivo em várias outras colunas (dia, dia da semana, mês, nome do mês, trimestre, e ano). Modelar o DW com vários campos de tempo é importante e consiste numa boa prática, pois permite que estes atributos sejam hierarquizados na modelagem de cubos. O segundo componente utiliza os campos construídos para construção de uma última coluna adicional, que consiste na data no formato DateTime. As colunas foram geradas utilizando a função SUBSTRING para obter campos específicos da *string* representando a data retornada pelo Twitter e estruturas condicionais.

5.3 Construção do Cubo OLAP

Após construção do processo de ETL previamente descrito, a última etapa antes de ser possível consultar o DW e construir aplicações analíticas é a construção do cubo OLAP. Para tal, foi utilizado o componente *Data Tools* do SQL Server. No ambiente, é possível construir modelos multidimensionais que interagem com dados de DW.

A construção de cubos no *Data Tools* é bastante simples. Ao criar um projeto do tipo *Analysis Services Multidimensional and Data Mining Project*, define-se a fonte de dados utilizada (no caso, o DW construído) e então é possível estruturar dimensões e criar cubos, a partir do assistente de criação de cubos. No assistente, basta informar qual a tabela-fato utilizada e quais dimensões serão associadas ao cubo. Por fim, também é possível selecionar colunas específicas de cada dimensão que serão acrescentados ao cubo e criar hierarquias para as mesmas.

Foi criado um único cubo para este estudo de caso, contendo todas as tabelas do DW modelado (*tweets* e suas dimensões: local, tempo e sentimento). Após seleção das dimensões do cubo, as colunas da tabela Dim_Date foram hierarquizados para permitir operações como *drill-down* e *roll-up*.

5.4 Consultas e Análises

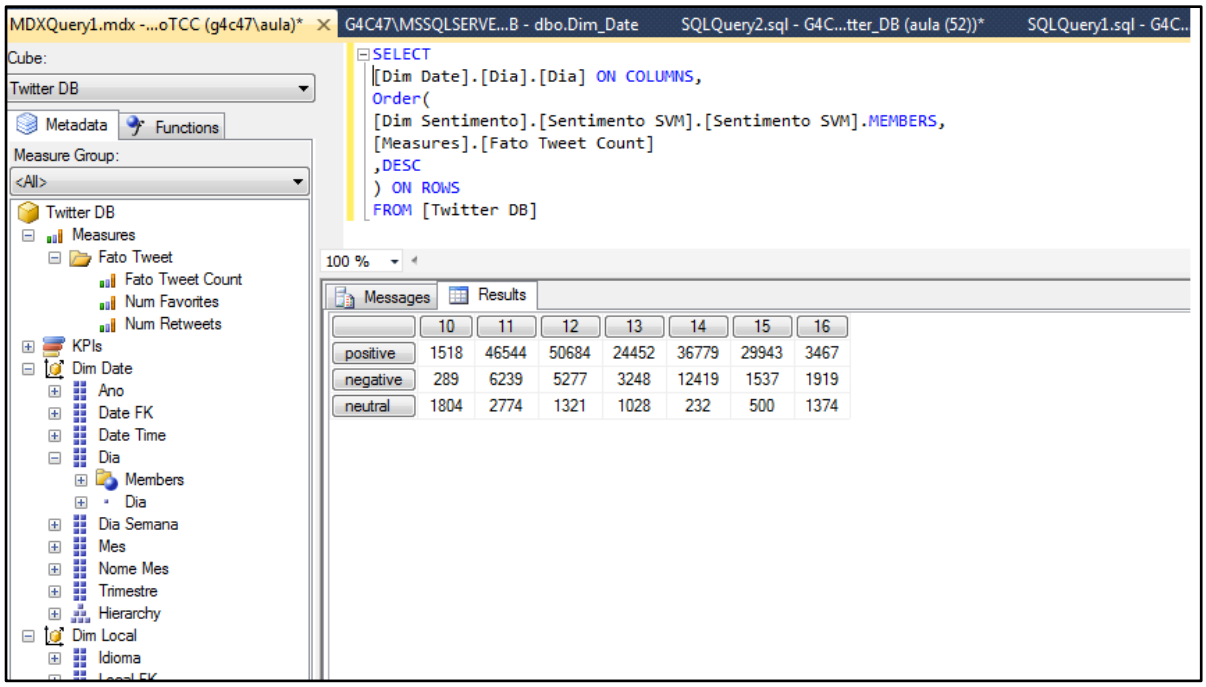
Há inúmeras formas de consultar e analisar dados multidimensionais de cubos OLAP. Atualmente, há dezenas de ferramentas de visualização disponíveis que normalmente têm como base operações sobre cubos OLAP (apresentadas anteriormente). Neste trabalho, foram exploradas duas formas de consultar e analisar os dados do cubo construído: por meio da utilização de uma linguagem de consulta, a MDX, e criação de tabelas dinâmicas.

5.4.1 Consultas em MDX

A linguagem MDX foi lançada pela Microsoft em 1997 como uma linguagem de consulta para bancos de dados OLAP, sendo equivalente à linguagem SQL para bancos de dados relacionais. Apesar de ser uma linguagem proprietária da Microsoft, atualmente MDX é amplamente adotada como linguagem padrão de consulta para ambientes OLAP, sendo usada por todos os grandes líderes do mercado de BI (MUNDY; THORNTHWAITE, 2011).

MDX provê uma sintaxe para consulta e manipulação de dados multidimensionais armazenados em cubos OLAP. Sua sintaxe é similar à de SQL, mas com algumas adaptações e funções nativas que permitem que sejam realizados cálculos com muito mais simplicidade. A Figura 16 mostra um exemplo de consulta construída para o DW apresentado, na qual se recupera a quantidade de *tweets* de cada classe de polaridade (neste exemplo, foi utilizada a máquina de vetores de suporte) por dia de monitoramento (10 a 16 de novembro de 2015).

Figura 16: Exemplo de consulta em MDX e conteúdo recuperado



Fonte: O Autor

5.4.2 Tabelas e Gráficos Dinâmicos

Tabelas dinâmicas (também conhecidas por tabelas-pivô ou *pivot tables*) são tabelas interativas normalmente utilizadas para consultar grandes quantidades de dados de forma amigável. Por meio deste tipo de estrutura, é possível realizar várias operações como agregações, expandir e recolher níveis de detalhamento, filtragens, classificações, entre outras. Estas operações são, de forma geral, diferentes combinações das operações sobre cubos OLAP apresentadas no Capítulo 4.

Para construção dos exemplos apresentados, foi utilizado o Microsoft Excel[50]. Apesar de amplamente utilizado e consagrado no mercado para criação de simples planilhas e gráficos, o Excel é uma ferramenta poderosa para interação com cubos OLAP. Atualmente, o software é a ferramenta de Business Intelligence mais utilizada no mundo (SALGADO; FONSECA; ÁVILA, 2015).

O Excel foi utilizado para estabelecer uma conexão com o cubo previamente construído. Com a conexão construída, o Excel mostra todas as métricas disponíveis (número total de tweets, número de curtidas, número de *retweets*) e todas as dimensões e seus atributos. A partir de então, é possível construir tabelas simplesmente selecionando quais campos deseja-se utilizar. A Figura 17 mostra um exemplo de tabela construída que sumariza a quantidade de *tweets*, por polaridade e por dia monitorado (mesmo conteúdo recuperado na consulta MDX mostrada na Figura 16), acrescentando-se o número de curtidas e *retweets* sobre estes *tweets*.

Figura 17: Exemplo de tabela dinâmica construída no Microsoft Excel

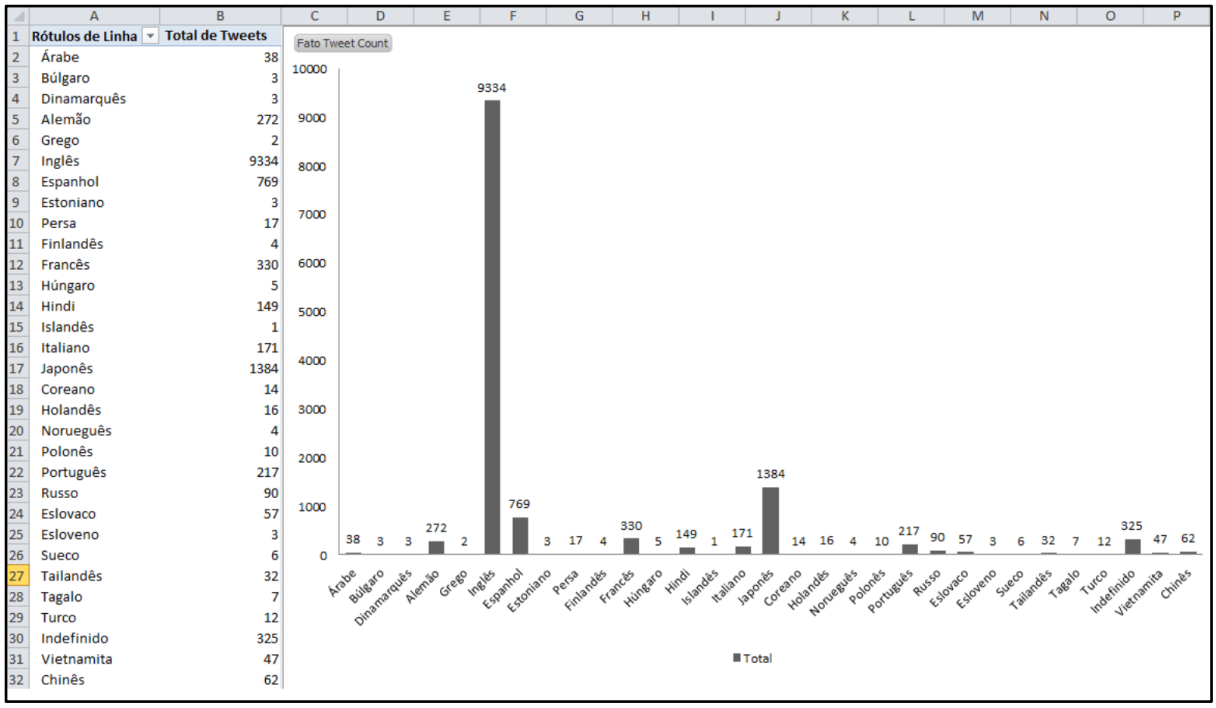
	A	B	C	D
1	Ano/Trimestre/Mês/Dia/Sentimento	Total de Tweets	Número de Curtidas	Número de Retweets
2	2015	13387	1184	233348
3	4	13387	1184	233348
4	11	13387	1184	233348
5	10	727	97	3611
6	Negativos	244	31	289
7	Neutros	108	0	1804
8	Positivos	375	66	1518
9	11	2861	495	55557
10	Negativos	916	121	6239
11	Neutros	258	3	2774
12	Positivos	1687	371	46544
13	12	3476	198	57282
14	Negativos	1467	75	5277
15	Neutros	195	15	1321
16	Positivos	1814	108	50684
17	13	2291	181	28728
18	Negativos	789	72	3248
19	Neutros	60	3	1028
20	Positivos	1442	106	24452
21	14	1352	66	49430
22	Negativos	633	37	12419
23	Neutros	34	1	232
24	Positivos	685	28	36779
25	15	1376	86	31980
26	Negativos	460	26	1537
27	Neutros	87	1	500
28	Positivos	829	59	29943
29	16	1304	61	6760
30	Negativos	656	29	1919
31	Neutros	82	1	1374
32	Positivos	566	31	3467
33	Total Geral	13387	1184	233348

Fonte: O Autor

Além de tabelas, é possível gerar gráficos dinâmicos sobre as tabelas construídas. O Excel permite personalização por meio da escolha dentre vários possíveis tipos de gráfico e *designs*. Na Figura 18, são mostrados um gráfico e a tabela correspondente, contendo informação sobre o número total de *tweets* por idioma.

[50] https://products.office.com/en-us/excel

Figura 18: Exemplo de tabela e gráficos dinâmicos construídos no Microsoft Excel



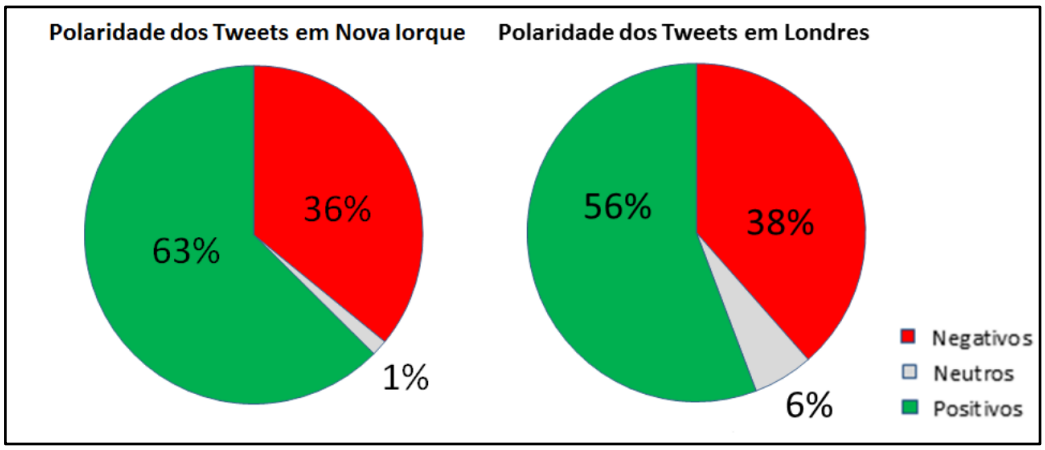
Rótulos de Linha	Total de Tweets
Árabe	38
Búlgaro	3
Dinamarquês	3
Alemão	272
Grego	2
Inglês	9334
Espanhol	769
Estoniano	3
Persa	17
Finlandês	4
Francês	330
Húngaro	5
Hindi	149
Islandês	1
Italiano	171
Japonês	1384
Coreano	14
Holandês	16
Norueguês	4
Polonês	10
Português	217
Russo	90
Eslovaco	57
Esloveno	3
Sueco	6
Tailandês	32
Tagalo	7
Turco	12
Indefinido	325
Vietnamita	47
Chinês	62

Fonte: O Autor

Tabelas dinâmicas são uma forma extremamente interessante de visualizar dados de um *data warehouse* pois, além de totalmente integradas e atualizadas automaticamente à medida que novos dados são incorporados ao DW, também permitem que os dados sejam manipulados com facilidade. É verdade que estruturas semelhantes poderiam ser construídas para bancos de dados transacionais ou até mesmo de arquivos de texto gerados na etapa de extração. Contudo, ao construir tabelas ou gráficos a partir de cubos OLAP, a manipulação dessas estruturas é feita com muita velocidade, sem atrasos relativos a processamento dos dados, devido à forma como os dados são agregados previamente nos cubos.

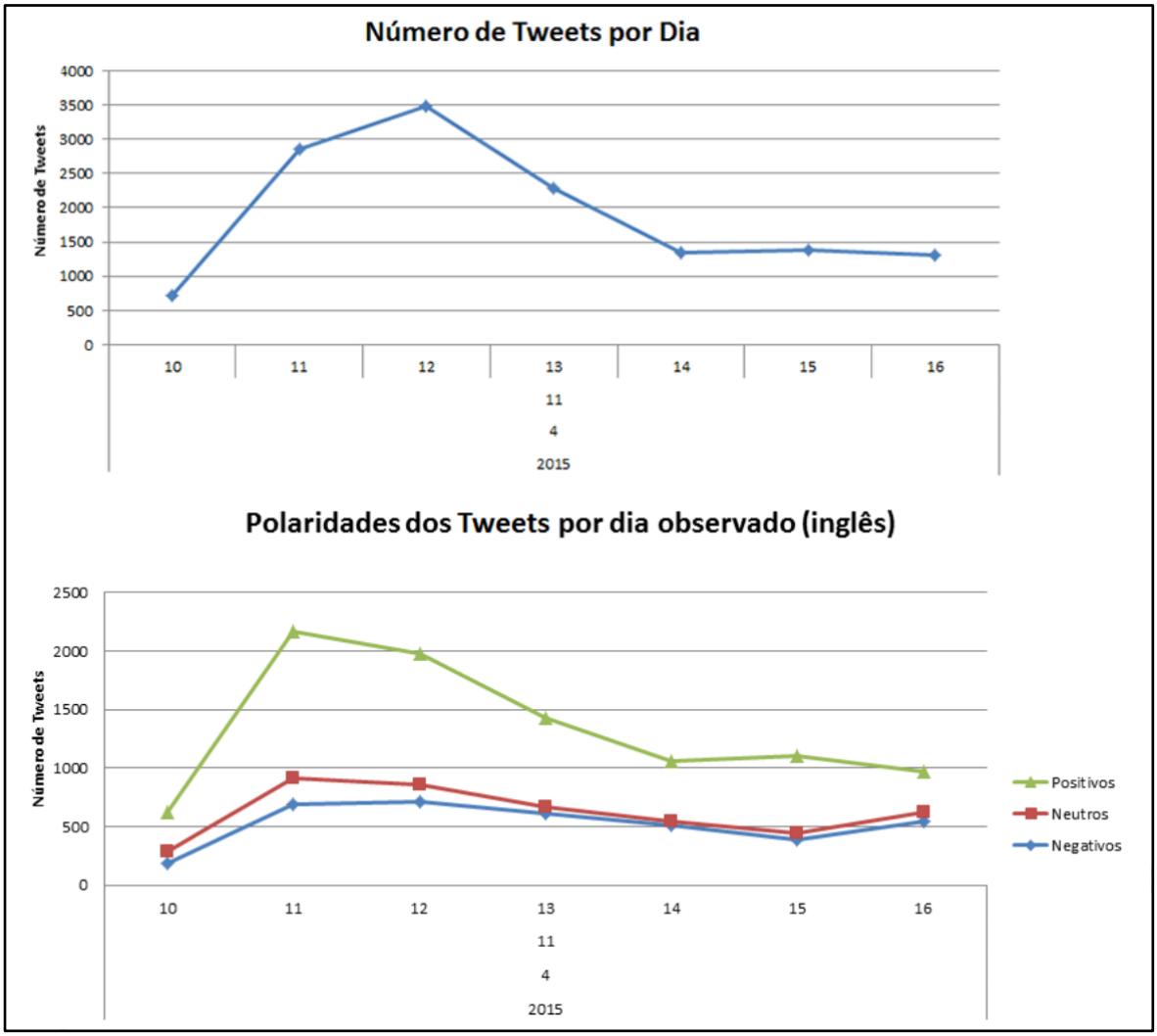
Sendo possível manipular dados e construir tabelas e gráficos em tempo real, são inúmeras as possibilidades de relatórios e análises que podem ser construídos e alterados com facilidade. Assim, o usuário de negócios pode se aproximar dos dados, sem a necessidade de linguagens de consulta para construir e responder perguntas importantes, como “Em que cidade o Surface Book tem melhor reputação? E pior?”, “Quais os idiomas mais falados por usuários do Surface Book?”, “Qual o dia em que mais se falou sobre o Surface Book?”, entre várias outras. As Figuras 19 e 20 mostram outros exemplos de gráficos dinâmicos, também construídos no Microsoft Excel.

Figura 19: Exemplo de gráficos dinâmicos mostrando a distribuição das classes de polaridades dos *tweets* em Nova Iorque e Londres, no período observado



Fonte: O Autor

Figura 20: Exemplo de gráficos dinâmicos mostrando o número de *tweets* publicados no período observado e a distribuição diária das classes de polaridade para os *tweets* em inglês



Fonte: O Autor

5.5 Considerações Finais

Após a implementação do estudo de caso apresentado, em que foi observado o funcionamento de toda a arquitetura proposta para dados do Twitter, ficou claro que bons resultados podem ser obtidos por meio da combinação de análise de sentimentos com *Business Intelligence*. Apesar do volume de dados utilizado ser relativamente pequeno (13 mil *tweets*), o sistema continuaria se comportando bem para volumes maiores, pelos argumentos já apresentados no Capítulo 4.

É importante ressaltar que foram vistos apenas alguns exemplos de formas de consultar e visualizar os dados. Contudo, em um contexto real, dezenas de ferramentas de visualização poderiam ser exploradas para investigar quais se adequam melhor às necessidades de cada organização. Com a quantidade de soluções disponíveis crescendo rapidamente, o poder das análises viabilizadas com a utilização de *data warehouses* também torna-se cada vez maior.

CAPÍTULO 6

Conclusão

O objetivo deste trabalho foi realizar um estudo sobre a viabilidade de uma arquitetura combinando dados extraídos de redes sociais online, algoritmos de mineração de opinião para classificação deste conteúdo e utilização de ferramentas de Business Intelligence para armazenamento e visualização da informação construída.

Inicialmente, foi explorado com profundidade o conceito de redes sociais, desde seu surgimento até o cenário atual. Neste contexto, foram investigadas as interfaces de programação disponibilizadas pelas principais plataformas sociais e construídas implementações que mostrassem o processo de extração de dados em cada uma delas. Após este estudo, concluiu-se que as redes sociais online tornaram-se parte da vida cotidiana de grande parte da população mundial e as estatísticas mostram que seu uso tende a aumentar. Esta grande penetração implica numa quantidade imensa de conteúdo que pode ser aproveitado para diversos tipos de análises. Apesar das limitações impostas pelas API de algumas redes sociais, este trabalho mostrou que extrair dados sociais é, de forma geral, totalmente possível e tipicamente bastante simples, sendo possível extrair grandes quantidades de dados com poucas linhas de código.

Em seguida, foi tratada da área de análise de sentimentos. Especificamente, a utilização de aprendizagem de máquina para classificar a polaridade de fragmentos de texto. Foram estudados diferentes classificadores e estratégias para pré-processar conteúdo textual. Após extenso embasamento teórico, mostrou-se por meio da classificação de dados extraídos do Twitter que a utilização de mineração de opinião para classificar conteúdo social é viável e pode atingir bons resultados. Foram comparados os desempenhos de três diferentes classificadores com diferentes configurações e concluiu-se que, apesar de todos atingirem taxas de acerto relativamente altas, a máquina de vetores de suporte obteve os melhores resultados, se mostrando uma boa alternativa para classificação de conteúdo social.

Por fim, foi realizado um estudo sobre a área de Business Intelligence. O trabalho mostrou detalhadamente o conceito de *Data Warehousing* e as vantagens de utilizar uma abordagem materializada com modelagem multidimensional para integrar, armazenar e consultar conteúdo social. A observação de que RSO têm crescido de forma assustadora nos últimos anos fortalece o argumento de utilização deste tibo de abordagem para lidar com dados extraídos de redes sociais, já que a principal motivação para utilização da área é sua capacidade de lidar com grandes quantidades de dados.

6.1 Contribuições

A principal contribuição deste trabalho foi mostrar como é possível combinar duas áreas extremamente poderosas, análise de sentimentos e *Business Intelligence*, para tratar de dados de redes sociais online, plataformas cuja utilização tem crescido muito rapidamente.

Assim, foi proposta uma arquitetura baseada na extração de dados destas plataformas, processamento por meio de técnicas de mineração de opinião e utilização de ferramentas BI para armazenamento e visualização.

Após investigação de cada uma das áreas individualmente, o trabalho confirmou a viabilidade de tal arquitetura por meio da apresentação de uma versão simplificada. Foi construído um sistema capaz de extrair publicações, classificá-las com relação às suas polaridades e carregar este conteúdo em um *Data Warehouse*. Por fim, foram apresentados exemplos de relatórios e análises potencialmente úteis para organizações que são viabilizadas com o uso desta arquitetura.

6.2 Trabalhos Futuros

Os resultados obtidos foram satisfatórios e suficientes para mostrar que a arquitetura proposta funcionaria bem em contextos reais de utilização. Contudo, pode-se realizar alguns trabalhos futuros para garantir resultados ainda melhores.

Primeiramente, pretende-se extrair dados não somente do Twitter, mas também de outras plataformas sociais. Para isso, será necessário adaptar a modelagem do *Data Warehouse* construída para que sua estrutura comporte dados de diferentes plataformas. Além disso, o módulo de extração de dados do processo ETL construído precisará ser expandido para que os dados das novas RSO possam ser extraídos.

No estudo de caso apresentado, todos os dados foram extraídos a partir do mesmo termo de consulta ("Surface book"). Também pretende-se tornar a arquitetura capaz de receber diferentes termos para monitoramento, por meio de uma interface gráfica com o usuário que modularize os dados de acordo com os termos dados.

Uma limitação enfrentada neste trabalho que deve ser corrigida no futuro é a forma como a informação sobre a localização dos dados extraídos do Twitter é tratada. A API disponibiliza apenas um campo textual contendo a localização, informada pelo usuário, mas sem padronização. Essa falta de padronização faz com que o sistema trate diferentes variações de um mesmo local como diferentes. Assim, será utilizada uma API de geolocalização (como o Bing Maps[51]) para conversão destes campos em coordenadas geográficas. Assim, além de padronizar os dados, será possível visualizar a distribuição das postagens em mapas.

Por fim, dispondo de um número maior de dados e de uma modelagem mais complexa do que a apresentada, existe a pretensão de realizar um estudo comparativo que mostre, por meio de métricas de desempenho, como arquiteturas OLAP têm desempenho superior para tratamento de largas quantidades de dados, quando comparadas com sistemas OLTP. Assim, ficará ainda mais evidente que o uso de *Business Intelligence* para o contexto social é uma excelente escolha.

[51] https://www.microsoft.com/maps/choose-your-bing-maps-API.aspx

APÊNDICE A

Implementação dos Classificadores

Figura 21: Implementação dos classificadores em Python 2.7

```python
import csv
import re
from nltk.stem import PorterStemmer
import nltk
from libsvm import svm
from libsvm import svmutil
from svmutil import *

def classToDouble(classificacao):
    return 0.0 if (classificacao == 'positive') else 1.0 if (classificacao == 'negative') else 2.0

def getMapaFeatures(features, featuresTweet):
    map = {}
    for w in features:
        map[w] = 0

    for feature in featuresTweet:
        map[feature] = 1
    return map

def gerarVetorFeaturesTreinamento(tweets, features):
    sortedFeatures = sorted(features)
    map = {}
    vetorFeatures = []
    classes = []
    for t in tweets:
        featuresTweet = t[0]
        classifTweet = t[1]
        map = getMapaFeatures(sortedFeatures, featuresTweet)
        vetorFeatures.append(map.values())
        classificacao = classToDouble(classifTweet)
        classes.append(classificacao)
    return {'vetorFeatures' : vetorFeatures, 'classes': classes}

def gerarVetorFeaturesTestes(tweets, features):
    sortedFeatures = sorted(features)
    map = {}
    vetorFeatures = []
    for tweet in tweets:
        map = getMapaFeatures(sortedFeatures, extrairFeatures(tweet, True))
        vetorFeatures.append(map.values())
    return vetorFeatures

def gerarMapaFeatures(tweet):
    featuresTweet = set(tweet)
    features = {}
    for word in listaFeatures:
        features['contains(%s)' % word] = (word in featuresTweet)
    return features

def similaridade(listaA, listaB):
    if (len(listaA) != len(listaB)):
        return 0.0
    equal = 0.0
    for i in range(len(listaA)):
        if (listaA[i] == listaB[i]):
            equal += 1.0
    return equal / float(len(listaA))

tweetsTreinamento = csv.reader(open('training.csv', 'rU'), delimiter=',', quotechar='"', dialect=csv.excel_tab)
stopWords = lerStopwords('stopwords.txt')
listaFeatures = []

tweets = []
for linha in tweetsTreinamento:
    sentimento = linha[0]
    tweet = linha[1]
    tweetPreprocessado = preprocessarTweet(tweet)
    vetorFeatures = extrairFeatures(tweetPreprocessado, stopWords)
    listaFeatures.extend(vetorFeatures)
    tweets.append((vetorFeatures, sentimento));

listaFeatures = list(set(listaFeatures)) # Remover ocorrencias repetidas
conjuntoTreinamento = nltk.classify.util.apply_features(gerarMapaFeatures, tweets)
resultado = gerarVetorFeaturesTreinamento(tweets, listaFeatures)
problem = svm_problem(resultado['classes'], resultado['vetorFeatures'])
param = svm_parameter('-q')
param.kernel_type = LINEAR
classifier = svm_train(problem, param)
```

```
classificadorSVM = svm_train(problem, param)
classificadorNB = nltk.NaiveBayesClassifier.train(conjuntoTreinamento)
classificadorME = nltk.classify.maxent.MaxentClassifier.train(conjuntoTreinamento, 'GIS', trace=3, \
                    encoding=None, labels=None, gaussian_prior_sigma=0, max_iter = 5)

conjuntoTestes = csv.reader(open('test.csv', 'rU'), delimiter=',', quotechar='"', dialect=csv.excel_tab)

listaTweetsTeste = []
classificacaoReal = []
classificacaoME = []
classificacaoNB = []
for teste in conjuntoTestes:
        tweet = teste[1]
        listaTweetsTeste.append(preprocessarTweet(tweet))
        classificacaoNaiveBayes = classificadorBayes.classify(gerarMapaFeatures(extrairFeatures(preprocessarTweet(tweet), True)))
        classificacaoMaxEntropy = classificadorME.classify(gerarMapaFeatures(extrairFeatures(preprocessarTweet(tweet), True)))
        classificacaoCorreta = teste[0]
        classificacaoReal.append(classToDouble(classificacaoCorreta))
        classificacaoNB.append(classToDouble(classificacaoNaiveBayes))
        classificacaoME.append(classToDouble(classificacaoMaxEntropy))
vetorFeaturesTestes = gerarVetorFeaturesTestes(listaTweetsTeste, listaFeatures)
p_labels, p_accs, p_vals = svm_predict([0] * len(vetorFeaturesTestes),vetorFeaturesTestes, classificadorSVM)

print "Naive Bayes - acuracia: ", (100.0*similaridade(classificacaoReal, classificacaoNB)), "%"
print "Max Entropy - acuracia: ", (100.0*similaridade(classificacaoReal, classificacaoME)), "%"
print "SVM - acuracia: ", (100.0*similaridade(classificacaoReal, p_labels)), "%"
```

Fonte: O Autor

Referências Bibliográficas

BAEZA-YATES, R., “Modern Information Retrieval: The Concepts and Technology behind Search” . 2. ed. Harlow: Addison-Wesley Professional, fev. 2011.

BENEVENUTO, F.; ALMEIDA, J.; SILVA, A., "Explorando Redes Sociais Online: Da Coleta e Análise de Grandes Bases de Dados às Aplicações", Mini-cursos do XXIX Simpósio Brasileiro de Redes de Computadores (SBRC). Campo Grande – MS, 2011.

BOIY, E.; MOENS, M., “A machine learning approach to sentiment analysis in multilingual Web texts”. Information Retrieval, vol. 12, n. 5, p. 526-558, 2009.

BOYD, D.; ELLISON, N., "Social Network Sites: Definition, History, and Scholarship". Journal of Computer-Mediated Communication, vol. 13, n. 1, p. 210-230, Wiley-Blackwell, 2007.

CHAUDHURI, S.; DAYAL, U., “An overview of data warehousing and OLAP technology”. ACM Sigmod Record, vol. 26, n. 1, p. 65-74, 1997.

COSTA, P. R. S.; SOUZA, F. F.; TIMES, V. C.; BENEVENUTO, F. “Towards Integrating Online Social Networks And Business Intelligence”, Proceedings of the International Conferences Web Based Communities and Social Media, p. 21-32. Lisboa, 2012.

DELLA PIETRA, S.; DELLA PIETRA, V.; LAFFERTY, J., “Inducing features of random fields”, IEEE Transactions on Pattern Analysis and Machine Intelligence, , vol. 19, n. 4, p. 380-393, 1997.

DOS ANJOS, F.; SAEGER, M.; DA SILVA, P., "Using Social Networks for Geo-Collaboration Through Oriented Architecture Service", 10th International Conference on Information Systems and Technology Management – CONTECSI. São Paulo – SP, 2013.

FACEBOOK, “Facebook Developers – The Graph API Documentation”, 2015. Disponível em: <https://developers.facebook.com/docs/graph-api/> [Acesso em 4 Set 2015].

FRANÇA, T. C.; FARIA, F. F.; RANGEL, R. M.; FARIAS, C. M.; OLIVEIRA, J., “Big Social Data: Princípios sobre Coleta, Tratamento e Análise de Dados Sociais”, XXIX Simpósio Brasileiro de Banco de Dados – SBBD ’14. Curitiba – PR, 2014.

FRIEDMAN, A., "1.2 billion smartphones sold in 2014, slowdown in growth seen for 2015", PhoneArena, 2015. Disponível em: <http://www.phonearena.com/news/1.2-billion-smartphones-sold-in-2014-slowdown-in-growth-seen-for-2015_id66085>. [Acesso em 31 Ago 2015].

GO, A.; BHAYANI, R.; HUANG, L., "Twitter Sentiment Classification using Distant Supervision", CS224N Project Report, p. 1–12, Stanford, 2009.

GOLDEN, B., “Amazon Web Services for Dummies”. 1. ed: John Wiley & Sons, 2013.

GOLFARELLI, M.; RIZZI, S.; CELLA, I., “Beyond data warehousing: what's next in business intelligence?”, Proceedings of the 7th ACM international workshop on Data warehousing and OLAP, p. 1-6, 2004.

GOOGLE, “Google+ Platform for Web – Google+ API”, 2015. Disponível em: <https://developers.google.com/+/web/api/rest/index> [Acesso em 4 Set 2015].

GUTIÉRREZ, A.; MAROTTA, A., “An overview of data warehouse design approaches and techniques”. Reportes Técnicos 01-09, 2001.

INMON, W., “Building the data warehouse”. 1. ed: John Wiley & Sons, 2005.

KIMBALL, R.; ROSS, M., “The data warehouse toolkit: the complete guide to dimensional modeling”. 2. ed: John Wiley & Sons, 2011.

KUNNEMAN, F.; LIEBRECHT, C.; VAN DEN BOSCH, A., "The (Un)Predictability of Emotional Hashtags in Twitter", Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM) – EACL 2014, p. 26-34, 2014.

LEWIS, D., “Naive (Bayes) at forty: The independence assumption in information retrieval.”, Machine learning: ECML-98, p. 4-15. Springer Berlin Heidelberg, 1998.

LUHN, H., "A business intelligence system", IBM Journal of Research and Development, v. 2, n. 4, p. 314-319, 1958.

MUNDY, J.; THORNTHWAITE, W., “The Microsoft data warehouse toolkit: With SQL Server 2008 R2 and the Microsoft Business Intelligence Toolset”. 2. ed: John Wiley & Sons, 2011.

NIGAM, K.; LAFFERTY, J.; MCCALLUM, A., “Using maximum entropy for text classification”, IJCAI-99 workshop on machine learning for information filtering, p. 61-67, 2009.

ORACLE, “Oracle9i Database Online Documentation”, 2002. Disponível em: <https://docs.oracle.com/cd/B10500_01/index.htm> [Acesso em 3 Nov 2015].

ORACLE, “Oracle Social Cloud: Social Relationship Management”, 2013. Disponível em: <http://www.oracle.com/us/products/social-relationship-mgmt-brief-1915605.pdf> [Acesso em 16 Nov 2015].

PANG, B.; LEE, S.; VAITHYANATHAN, S., “Thumbs up? Sentiment classification using machine learning techniques", Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), p. 79-86, 2002.

PANG, B.; LEE, L., “Opinion mining and sentiment analysis: Foundations and Trends in Information Retrieval”, vol. 2, n. 1-2, p. 1-135, 2008.

POLONI, K. M.; TOMAÉL, K. I., “Coleta de Dados em Plataformas de Redes Sociais: Estudo de Aplicativos”, Anais do III Workshop de Pesquisa em Ciência da Informação – III WPCI ‘14. Londrina - PR, 2014.

PORTER, M. F., “An algorithm for suffix stripping”, Program, vol. 14, n. 3, p. 130-137. Cambridge, 1980.

POTTS, C., “Sentiment Symposium Tutorial”, Sentiment Analysis Symposium – Stanford Linguistics. São Francisco, 2011.

POWER, D., “Decision support systems: a historical overview”. Handbook on Decision Support Systems 1. Springer Berlin Heidelberg, p. 121-140, 2008.

RUSSELL, M. A., “Mining the Social Web” . 2. ed. Sebastopol, CA: O’Reilly, out. 2013.

RUBEN CUEVAS, R. G.; REZA REJAIE, R. M.; CUEVAS, A., "Google+ or Google-? Dissecting the Evolution of the New OSN in its First Year", Proceedings of WWW – World Wide Web Conference, Rio de Janeiro, 2013.

SALGADO, A. C.; FONSECA, F., ÁVILA, F., “Data Warehousing e BI”, 2015. Disponível em: <http://cin.ufpe.br/~if696/aulas/DW_Aula_1.pdf> [Acesso em 3 Nov 2015]

SALGADO, A. C.; LÓSCIO, B., “Integração de dados na Web”. Escola Regional De Informática Da SBC – Regional De São Paulo, vol. 6, p. 157-174, 2001.

SALLAM, R. L.; HOSTMANN, B.; SCHLEGEL, S.; TAPADINHAS, J.; PARENTAU, J., “Gartner Magic Quadrant for Business Intelligence Platforms – 2015”. [Online] Disponível em: <http://www.gartner.com/technology/reprints.do?id=1-2ACLP1P&ct=150220&st=sb> [Acesso em 26 Jul 2015].

STEMPEL, G. H.; HARGROVE, T.; BERNT, J. P., “Relation of Growth of Use of the Internet to Changes in Media Use from 1995 to 1999”, Journalism & Mass Communication Quarterly, SAGE Journals, vol. 77, n. 1, p. 71-79, set. 2000.

TONG, S.; KOLLER, D., “Support vector machine active learning with applications to text classification”, The Journal of Machine Learning Research, vol. 2, p. 45-66, 2002.

TWITTER, “Developer Documentation”, 2015. Disponível em: <https://dev.twitter.com/> [Acesso em 4 Set 2015].

WASSERMAN, S.; FAUST, K., "Social Network Analysis in the Social and Behavioral Sciences". Social Network Analysis: Methods and Applications. Cambridge University Press. p. 1–27. Cambridge, 1994.

WATSON, H.; WIXOM, B., “The current state of business intelligence”, Computer, v. 40, n. 9, p. 96-99, 2007.

WEBER, L., “Marketing to the Social Web: How Digital Customer Communities Build Your Business”. 2. ed. Hoboken: John Wiley & Sons, out. 2009.