UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

EVERTON SCHNEIDER DOS SANTOS

RECOMENDAÇÃO DE CONTEÚDO EM UM CONTEXTO DE BIG DATA

TRABALHO DE CONCLUSÃO DE CURSO

MEDIANEIRA
2015

EVERTON SCHNEIDER DOS SANTOS

RECOMENDAÇÃO DE CONTEÚDO EM UM CONTEXTO DE BIG DATA

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina de Trabalho de Conclusão de Curso II, do Curso Superior de Bacharelado em Ciência da Computação do Departamento Acadêmico de Computação – DACOM – da Universidade Tecnológica Federal do Paraná - UTFPR, como requisito parcial para obtenção do título de Bacharel.

Orientador: Prof. Arnaldo Candido Junior, Dr.

MEDIANEIRA

2015

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Departamento Acadêmico de Computação
Curso de Ciência da Computação

TERMO DE APROVAÇÃO

RECOMENDAÇÃO DE CONTEÚDO EM UM CONTEXTO DE BIG DATA

Por

EVERTON SCHNEIDER DOS SANTOS

Este Trabalho de Conclusão de Curso (TCC) foi apresentado às 13:30 h do dia 12 de junho de 2015 como requisito parcial para a obtenção do título de Bacharel no Curso Superior de Bacharelado em Ciência da Computação, da Universidade Tecnológica Federal do Paraná, *Câmpus* Medianeira. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Arnaldo Candido Junior
UTFPR – *Câmpus* Medianeira
(Orientador)

Prof. Dr Claudio Leones Bazzi
UTFPR – *Câmpus* Medianeira
(Convidado)

Prof. Dr. Pedro Luiz de Paula Filho
UTFPR – *Câmpus* Medianeira
(Convidado)

Prof. MSc Jorge Aikes Junior
UTFPR – *Câmpus* Medianeira
(Convidado)

Prof. MSc Juliano Rodrigo Lamb
UTFPR – *Câmpus* Medianeira
(Responsável pelas atividades de TCC)

A folha de aprovação assinada encontra-se na Coordenação do Curso.

AGRADECIMENTOS

Agradeço ao professor Jean Metz, pela ajuda inestimável na criação desse trabalho. Também gostaria de agradecer ao professor Arnaldo Candido Junior por ter aceitado o desafio de continuar e finalizar esse trabalho. Gostaria de agradecer o professor Juliano Rodrigo Lamb pelo apoio na formatação deste trabalho, desde sua concepção até a finalização. Por fim, obrigado aos meus colegas e familiares pelo apoio incondicional.

RESUMO

SANTOS, Everton Schneider dos. Recomendação de conteúdo em um contexto de *Big Data*. 2015. 49 f. Trabalho de Conclusão do Curso Superior de Bacharelado em Ciência da Computação. Universidade Tecnológica Federal do Paraná. Medianeira, 2015.

Este trabalho propõe o uso de recomendação de conteúdo para extração de conhecimento em redes sociais. O processo é amparado nas abordagens de *Big Data*, além de fazer uso de mineração de sentimentos. O crescimento na geração de conteúdo em redes sociais é um desafio para empresas e pesquisadores que desejam extrair conhecimento em tempo real de volumes massivos de dados. Avaliou-se a preferência dos usuários por empresas de tecnologia. Para medir a preferência dos usuários por um determinado item foi utilizada a abordagem baseada em mineração de sentimentos. Os dados gerados foram então utilizados para construir recomendadores de conteúdo e os resultados obtidos foram avaliados. O recomendador do tipo *Userbased*, com a vizinhança do tipo *NearestNUser* e a medida de similaridade *Tanimoto Coefficient* obteve os resultados que mais se aproximaram do ideal, com um *score* de 0.020, precisão de 0.246, cobertura de 0.259 e medida F1 de 0.252.

Palavras-chaves: Aprendizagem de máquina, *Mahout*, mineração de sentimentos, *Liwc*, *Twitter*.

ABSTRACT

SANTOS, Everton Schneider dos. Content Recommendation on a *Big Data* context. 2015. 49 f. Trabalho de Conclusão do Curso Superior de Bacharelado em Ciência da Computação. Universidade Tecnológica Federal do Paraná. Medianeira, 2015.

This work uses knowledge extraction in social media based on content recommendation. The process is aided in Big Data aproaches and uses sentiment mining. The growing in social media is a challenge for companies and reasearchers focusing knowledge extraction of massive data in real time. This work evaluated the user preference for technology companies. Sentiment mining is used to measure user preference for a given company. The gathered data are used to build content recommenders and its results are analyzed. The UserBased recommender, with NearestNUser neighborhood and Tanimoto Coefficient similary obtained results closest to ideal, with a score of 0.020, precision of 0.246, recall of 0.259 and F1 measure of 0.252.

Keywords: Machine Learning, Mahout, opinion mining, Big Data, Liwc, Twitter.

LISTA DE FIGURAS

Figura 1 - Recomendação de itens da Amazon 18
Figura 2 - Recomendação Baseada no Usuário.....27
Figura 3 - Recomendação Baseada no Item.....29
Figura 4 - Diagrama de casos de uso 34
Figura 5 - Diagrama de classes: Pacote Extração 35
Figura 6 - Diagrama de classes: Pacote Pré-processamento 35
Figura 7 - Diagrama de classes: Pacote Recomendação 36
Figura 8 - Visão geral do Sistema 37
Figura 9 - Extração e armazenamento dos dados.....38
Figura 10 - Pré-processamento 1.....39
Figura 11 - Pré-processamento 2.....40
Figura 12 - Modelo do arquivo de preferências.....41

LISTA DE TABELAS

Tabela 1 - Resultado da avaliação do recomendador UserBased utilizando vizinhança do tipo NearestNUser 43
Tabela 2 - Resultado da avaliação do recomendador UserBased utilizando vizinhança do tipo ThresholdUser 44
Tabela 3 - Resultado da avaliação do recomendador ItemBased 45
Tabela 4 - Resultado da avaliação dos recomendadores que não utilizam medida de similaridade e vizinhança 45

SUMÁRIO

1 INTRODUÇÃO 11
1.1 OBJETIVO GERAL 11
1.2 OBJETIVOS ESPECÍFICOS 11
1.3 JUSTIFICATIVA 12
1.4 ORGANIZAÇÃO DO TRABALHO 13
2 RECOMENDAÇÃO DE CONTEÚDO 15
2.1 APRENDIZADO DE MÁQUINA 15
2.2 MINERAÇÃO DE SENTIMENTOS 16
2.3 SISTEMAS DE RECOMENDAÇÃO 17
3 BIG DATA 20
3.1 CARACTERÍSTICAS 20
3.2 BANCO DE DADOS NÃO RELACIONAL 21
3.3 ANÁLISE EM BIG DATA 23
3.4 BIG DATA E REDES SOCIAIS 23
4 FERRAMENTAS PARA BIG DATA 25
4.1 APACHE HADOOP 25
4.2 APACHE MAHOUT 26
4.2.1 Recomendação de Conteúdo 26
4.2.2 Recomendação Baseada no Usuário 27
4.2.3 Recomendação Baseada no Item 28
4.2.4 Medidas de Similaridade 29
4.2.5 Avaliação de Recomendadores 31
5 MATERIAL E MÉTODOS 33
5.1 ARQUITETURA DO SISTEMA 33
5.1.1 Diagrama de Classe 34
5.2 IMPLEMENTAÇÃO 36
5.2.1 Extração e armazenamento dos dados 37
5.2.2 Mineração de sentimentos e pré-processamento dos dados 39
5.2.3 Recomendação 41

6 RESULTADOS OBTIDOS 43
6.1 USERBASEDRECOMMENDER 43
6.2 ITEMBASEDRECOMMENDER 44
6.3 OUTROS RECOMENDADORES 45
6.4 RESULTADO DAS RECOMENDAÇÕES 46
7 CONCLUSÃO 47
7.1 TRABALHOS FUTUROS 47
REFERÊNCIAS 48
APÊNDICE A – VISÃO GERAL DO DIAGRAMA DE CLASSES 51

1 INTRODUÇÃO

O crescimento no número de usuários de redes sociais, aliado ao custo cada vez menor de armazenamento de dados trouxe novos desafios para empresas e pesquisadores na área de análise e mineração de dados, como o significativo aumento no volume de dados produzidos em uma velocidade nunca antes observada, além do surgimento de novas fontes de dados heterogêneos.

Na tentativa de enfrentar esses desafios, ferramentas como o *Hadoop*[1] e a biblioteca *Mahout*[2] vem ganhando notoriedade no desenvolvimento de soluções cada vez melhores para os problemas criados em função desse grande volume de dados.

Nesse contexto, a biblioteca *Apache Mahout* ganha destaque ao aplicar técnicas de inteligência artificial na busca por soluções cada vez melhores para os problemas de *Big Data*.

1.1 OBJETIVO GERAL

O objetivo deste trabalho é desenvolver um mecanismo para a recomendação de postagens patrocinadas para redes sociais, em um contexto de *Big Data*, utilizando a biblioteca *Apache Mahout* aplicada a técnicas de mineração de sentimentos.

1.2 OBJETIVOS ESPECÍFICOS

São objetivos específicos deste trabalho:

a. Modelar e desenvolver uma solução computacional para a extração de dados de redes sociais;
b. Modelar e desenvolver soluções para o pré-processamento dos dados extraídos, adequando-os para a leitura;

[1] https://hadoop.apache.org/
[2] http://mahout.apache.org/

c. Modelar e desenvolver um componente de recomendação baseado em mineração de sentimentos utilizando a biblioteca *Apache Mahout*;
d. Dividir os dados e criar dos conjuntos de teste e validação;
e. Avaliar os resultados provenientes da utilização do recomendador nos dados previamente extraídos e processados.

1.3 JUSTIFICATIVA

O “dilúvio de dados” é o nome dado ao fenômeno de grande aumento na geração de informação em negócios, governo e ciência (TAN et al., 2013). Ainda segundo Tan et. al. (2013), esse crescimento foi responsável por afetar profundamente a forma como as pessoas processam e interpretam novos conhecimentos. Como a maior parte desses dados se origina e reside na Internet, um desafio evidente é determinar como tecnologia de computação na Internet deve evoluir para nos permitir acessar, reunir, analisar e agir em face ao *Big Data*.

Ao analisar a relevância do fenômeno *Big Data* em como os dados de redes sociais são vistos e tratados, Tufekci (2014) indica que, o surgimento de *Big Data* de redes sociais teve impactos no estudo do comportamento humano similares ao da introdução do microscópio ou do telescópio nos campos da biologia e astronomia, produziu uma mudança qualitativa na escala, escopo e profundidade de uma possível análise.

Porém, o aumento na quantidade de dados gerados em redes sociais traz novos desafios para os profissionais envolvidos no estudo e utilização de *Big Data*, como a manipulação e processamento de consultas, além de desafios na área de base de dados e mineração de dados (VIEIRA et al., 2012).

É nesse contexto, de buscar novas e melhores soluções para os problemas decorrentes do crescimento na geração de dados nos mais diversos campos, que o *Apache Hadoop* e a biblioteca *Apache Mahout* ganham destaque. Dumbill, et al. (2012) apontam que o *Hadoop* vem sendo a força propulsora por trás do crescimento da indústria de *Big Data*, já que o *Hadoop* é capaz de processar grandes quantidades de dados de forma barata e independente de sua estrutura.

No cenário de ferramentas e bibliotecas que formam o *Apache Hadoop*, destaca-se a biblioteca *Mahout*, um projeto ainda em desenvolvimento que vem sendo usado principalmente em motores de recomendação, classificadores de documentos e para resolver problemas de predição (ESTEVES; RONG, 2011). Por serem projetos de código aberto, *Hadoop* e *Mahout* se destacam por seu baixo custo de implementação e, segundo Esteves e Rong (2011), podem ser usados como uma solução econômica para resolver problemas de aprendizagem de máquina.

Desta maneira, a vasta quantidade de dados disponíveis em formato digital, juntamente com grupos de usuários maiores e bem organizados, facilitam uma melhoria significativa em inteligência e conhecimento sociais derivados de dados públicos. Isso pode ser resumido como uma sobreposição de redes sociais para análises de *Big Data*. Essa área apresenta uma riqueza de novas oportunidades de pesquisa para engenheiros e cientistas (TAN et. al., 2013).

A análise dos conceitos relativos à redes sociais e *Big Data* revela a importância do estudo dessas áreas e a forma como elas se relacionam já que, segundo Tufekci (2014), a *social media big data* tem sido um elemento chave para percepções cruciais em relação ao comportamento humano, além de permitir a observação de fenômenos sociais em um nível jamais visto anteriormente.

Tendo em vista a importância do estudo de redes sociais e *Big Data* e, levando em consideração que a mineração de dados em redes sociais é um campo de estudo emergente no qual existem mais problemas do que soluções prontas (ZARAFANI; ABASSI; LIU, 2014), a proposta desse trabalho é estudar as ferramentas e conceitos de aprendizagem de máquina citados anteriormente para auxiliar a busca por novas e melhores soluções em *Big Data* e redes sociais.

1.4 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado da seguinte maneira: no capítulo 2 são definidos os principais conceitos relacionados a Recomendação de Conteúdo, como o aprendizado de máquina e a mineração de sentimentos. O capítulo 3 traz o conceito de *Big Data*, sua definição, a exploração de suas características, a utilização de bancos de dados não relacionais, como é feita a análise em *Big Data* e sua relação

com redes sociais. O capítulo 4 explora as ferramentas para *Big Data*, especialmente o *Apache Hadoop* e o *Apache Mahout*. O capítulo 4.2, que trata do *Mahout*, foca na recomendação de conteúdo, especialmente as baseadas no usuário e no item, nas medidas de similaridade e na avaliação de recomendadores. Por fim, o material e métodos do trabalho são explicados no capítulo 5, o capítulo 6 discute os resultados obtidos e o trabalho termina no capítulo 7 com a conclusão e uma breve discussão de trabalhos futuros.

2 RECOMENDAÇÃO DE CONTEÚDO

Nesta seção são apresentados os principais conceitos e teorias relacionados à recomendação de conteúdo e aos diferentes tipos de sistemas de recomendação.

2.1 APRENDIZADO DE MÁQUINA

A área de Inteligência Artificial já foi vista como uma área teórica, aplicável à problemas como pouco valor prático (FACELI et al., 2011). Porém, na década de 1970 houve uma disseminação no uso de técnicas de Inteligência Artificial para a solução de problemas reais. A aquisição de conhecimento era feita através de entrevistas com especialistas do domínio. Esse processo era limitado por sua subjetividade, já que a tomada de decisão dependia da intuição do especialista.

O crescimento da complexidade dos problemas e do volume de dados gerados pelas mais diferentes áreas evidenciou a necessidade de novas ferramentas, mais sofisticadas e autônomas, que dependiam menos de especialistas. Essas novas técnicas deveriam ser capazes de, utilizando experiências passadas, criar uma hipótese ou função capaz de resolver um determinado problema. O processo de utilização das experiências passadas na indução de uma hipótese é conhecido como Aprendizado de Máquina (FACELI et. al., 2011).

Aprendizado de máquina é o processo pelo qual um sistema de computação se torna melhor na solução de um problema com base na experiência, ou seja, a solução de problemas anteriores (MITCHELL, 1997). Diferentes algoritmos se adéquam a diferentes questões. Classificação, análise de regressão e sistemas de recomendação são três das mais conhecidas técnicas de aprendizado de máquina (ANDERSSON; BOGREN; BREDMAR, 2014).

A classificação tem como objetivo conectar novas observações às classes com a ajuda de dados de treinamento prévio. O algoritmo de classificação, chamado de indutor, pode, por exemplo, usar dados meteorológicos para prever uma tempestade, baseado em observações passadas (ANDERSSON; BOGREN; BREDMAR, 2014).

Predição do preço de ações ou vendas futuras são problemas que sempre interessaram, e ainda interessam as mais diversas organizações. As técnicas implementadas pela análise de regressão tentam solucionar o problema representando predições na forma de funções matemáticas. Essas técnicas calculam uma função, ou curva, aplicada aos dados de entrada. Um exemplo de utilização da análise de regressão é a utilização de dados históricos para predizer a temperatura de uma cidade (ANDERSSON; BOGREN; BREDMAR, 2014).

2.2 MINERAÇÃO DE SENTIMENTOS

As palavras que as pessoas usam diariamente refletem suas visões de mundo e suas relações sociais. A linguagem é a forma mais comum e confiável para indivíduos traduzirem seus pensamentos e emoções para uma forma que outros possam entender. O desenvolvimento simultâneo de computadores pessoais muito rápidos, Internet e a criação de novas estratégias estatísticas ajudaram a criar uma nova era do estudo psicológico da linguagem. Recorrendo à grandes quantidades de texto, pesquisadores podem ligar o uso da linguagem informal com medidas comportamentais e auto-referidas de personalidade, comportamento social e estilos cognitivos (TAUSCZIK; PENNEBAKER, 2010).

A mineração, ou análise, de sentimentos é definida como o processo de detectar automaticamente se um segmento de texto possui conteúdo opinativo ou emocional e extrair sua polaridade ou valência (PALTOGLOU et al., 2010). Esse é um tópico de pesquisa relativamente novo no processamento de linguagem natural e que ganhou muita atenção devido ao crescimento da Internet social. Uma tarefa comum da mineração de sentimentos é a classificação de textos. Um texto ou sentença pode ser classificado como positivo, negativo ou neutro (BALAGE; PARDO; ALUÍSIO, 2013).

O aumento na geração de conteúdo na internet resultou em uma riqueza de informação que é de grande importância para instituições e empresas, proporcionando-lhes dados para pesquisar seus consumidores, gerenciar suas reputações e identificar novas oportunidades (PALTOGLOU et al., 2010).

A mineração de sentimento pode ser feita utilizando a abordagem de aprendizado de máquina, com um conjunto de características que são aprendidas de um *corpus* anotado ou de exemplos rotulados. Um outro tipo de abordagem utiliza um léxico que fornece a polaridade, ou orientação semântica, para cada palavra ou frase no texto. O principal componente dessa abordagem é o recurso léxico, que deve ser preciso e ter uma boa cobertura de vocabulário (BALAGE; PARDO; ALUÍSIO 2013).

A abordagem baseada em léxico não necessita de um *corpus* anotado, e é conhecida por sua independência de domínio, enquanto a abordagem de aprendizado de máquina tende a adaptar ao domínio que o classificador foi treinado (AUE; GAMON, 2005).

Um léxico compreensivo e de alta qualidade é muitas vezes essencial para uma mineração de sentimentos em larga escala de forma rápida e precisa. Um exemplo de léxico que tem sido amplamente utilizado no domínio de redes sociais é o *Linguistic Inquiry and Word Count* (LIWC). Sociólogos, psicólogos, linguistas e cientistas da computação acham o LIWC atrativo pois ele tem sido extensivamente validado. Além disso, seu dicionário direto e listas de palavras simples são facilmente inspecionados, entendidos e estendidos se necessário (HUTTO; GILBERT, 2014).

2.3 SISTEMAS DE RECOMENDAÇÃO

Sistemas de recomendação são muito utilizados na recomendação de produtos. Seu objetivo é recomendar itens que sejam do interesse de um indivíduo. Em algumas situações, os dados estão disponíveis mas são incompletos, com valores ausentes. Em determinadas situações, isso é algo intencional e não uma falha dos dados. Um sistema de recomendação tenta preencher esses valores ausentes usando conhecimento corrente. Em outras palavras, novos valores são recomendados. Recomendação baseada em modelo e recomendação baseada e vizinhança são duas abordagens que tentam preencher essas lacunas (ANDERSSON; BOGREN; BREDMAR, 2014).

Um exemplo prático da utilização de técnicas de recomendação pode ser visto na Figura 1, que mostra o sistema de recomendação usado pela *Amazon*. Outras

empresas de vendas *online* também utilizam sistemas parecidos para recomendar produtos a seus clientes.

Figura 1 - Recomendação de itens da Amazon
Fonte: Autoria Própria

O algoritmo de recomendação aprende uma função que determina um valor real para cada par ordenado item-usuário (*u, i*), esse valor indica o grau de interesse do usuário *u* em relação ao item *i*, e denota a avaliação dada pelo usuário *u* ao item *i*. O algoritmo de recomendação não está limitado a recomendação de itens, podendo ser generalizado para a recomendação de propaganda ou conteúdo (ZAFARANI; ABBASI; LIU, 2014).

Entre as principais formas de recomendação podem ser citadas:

- Recomendação Baseada em Vizinhança - nessa abordagem, os vizinhos decidem qual deve ser o valor do atributo ausente do usuário. Uma técnica comum quando se procura por vizinhos é usar o conhecimento de domínio específico sobre o problema. Nesse caso, todos os vizinhos devem ter a mesma influência na hora de escolher um valor (ANDERSSON; BOGREN; BREDMAR, 2014).

  Sistemas de recomendação baseados em vizinhança são fáceis de entender e implementar mas possuem falhas ao serem aplicados em um contexto de *Big Data*. Todo o conjunto de dados deve ser considerado a fim de encontrar vizinhos toda vez que uma predição é feita, logo, o conjunto de dados deve estar carregado em memória. Mesmo com esse problema, recomendadores baseados em vizinhança são amplamente utilizados (ANDERSSON; BOGREN; BREDMAR, 2014).

- Recomendação Baseada em Conteúdo - sistemas de recomendação baseados em conteúdo se apoiam no fato de que o interesse de um usuário deve corresponder a descrição dos itens recomendados pelo sistema. Ou seja, quanto mais similar é a descrição de um item em relação ao interesse do usuário, maior é a chance de ele achar essa recomendação interessante. Recomendadores implementam a ideia de medir a similaridade entre a descrição de um item e a informação do perfil de um usuário. Quanto maior a similaridade, maior a chance de um item ser recomendado (ZAFARANI; ABBASI; LIU, 2014).
- Filtragem Colaborativa - Na filtragem colaborativa, um conjunto de técnicas de recomendação clássicas, normalmente é dada uma matriz item-usuário, na qual cada entrada é desconhecida ou é o valor de avaliação dado por um usuário a um item. O objetivo da filtragem colaborativa é predizer as avaliações ausentes e possivelmente recomendar o item com a maior avaliação predita para o usuário. Essa predição pode ser executada diretamente, usando avaliações anteriores na matriz (ZAFARANI; ABBASI; LIU, 2014).

  Essa abordagem é chamada de filtragem colaborativa baseada em memória, pois utilizada dados históricos disponíveis na matriz. Alternativamente, pode-se supor que um modelo fundamental (hipótese) governa a maneira que os usuários avaliam os itens. Esse modelo pode ser aproximado e aprendido. Após ser aprendido, esse modelo pode ser usado para prever outras avaliações. Essa segunda abordagem é conhecida como filtragem colaborativa baseada em modelo (ZAFARANI; ABBASI; LIU, 2014).

3 BIG DATA

Big Data pode ser definida como uma área de pesquisa sobre grandes quantidades de dados que necessita de novas tecnologias e arquiteturas para que possa ser possível extrair valor desse conjunto de dados, já que o tamanho desse conjunto dificulta a análise dos dados por meio de técnicas tradicionais (KATAL; WAZID; GOUDAR, 2013).

Os dados que organizações e pesquisadores podem armazenar e analisar são divididos em dois tipos: os estruturados e armazenados em bancos de dados relacionais tradicionais e os dados não estruturados, como fotos, vídeos ou mesmo gerados por máquinas como sensores (HURWITZ et al., 2013). A massiva quantidade de novos tipos de dados gerados traz também novos tipos de desafios para indivíduos e empresas que desejam extrair algum tipo de valor ou conhecimento desses dados.

Hurwitz et al.(2013) complementa essa definição ao afirmar que *Big Data* é a capacidade de gerenciar um grande volume de dados diferentes, na velocidade correta e no tempo correto para permitir reação e análises em tempo real.

3.1 CARACTERÍSTICAS

Katal, Wazid e Goudar (2013) citam que o termo *Big Data*, atualmente, é utilizado de maneira imprópria já que foca apenas no volume dos dados, não dando a atenção devida para suas outras propriedades. As propriedades, ou dimensões, que diferenciam o *Big Data* das formas tradicionais de armazenamento e análise de dados são:

- Volume – O tamanho do conjunto de dados, que é muito maior se comparado as formas tradicionais de dados (LIU; YANG; ZHANG, 2013). Não só a quantidade, segundo Dong & Srivastava (2013), o número de fontes a partir das quais é possível extrair dados é muito maior em *Big Data* do que em contextos tradicionais de armazenamento;
- Velocidade – O quão rápido os dados são criados e armazenados. McAfee & Brynjolfsson (2012) citam que a velocidade na criação dos dados pode

ser mais importante para algumas aplicações do que o volume. Novas fontes, como redes sociais, produzem dados em tempo real. Outras, como sensores, produzem dados que estão em constante movimentação no banco de dados, que são impossíveis de analisar utilizando sistemas tradicionais (KATAL; WAZID; GOUDAR, 2013);

- Variedade – Destaca a heterogeneidade dos dados armazenados. Nem todos os dados extraídos são estruturados. Os conjuntos de dados também podem ser semiestruturados ou não estruturados. Zhang e Huang (2013) apontam para a dificuldade que gerenciadores de bancos de dados tradicionais enfrentam diante da manipulação de dados não estruturados, já que a quantidade de informação, gerada pelas centenas de atributos em múltiplas dimensões do conjunto, supera a capacidade de manipulação e análise dos sistemas tradicionais.

Essa representação em três dimensões, apesar de conveniente, pode ser muito simplista, já que é possível o processamento de um pequeno conjunto de dados complexos ou um grande conjunto de dados simples. Por isso existem outras propriedades tão importantes quanto as já citadas, como o *valor* e a *veracidade* dos dados. A veracidade dos dados define qual o grau de precisão dos dados em predizer valor de negócio (HURWITZ et al., 2013).

3.2 BANCO DE DADOS NÃO RELACIONAL

As propriedades que definem *Big Data* foram responsáveis por criarem novos desafios relacionados a forma como os dados são armazenados em ferramentas tradicionais. Entre os desafios encontrados, Vieira et al. (2012) citam a necessidade de maior poder de processamento, suporte a dados heterogêneos e a dificuldade na modelagem de tais dados. Esses e outros desafios influenciaram diretamente na criação de sistemas de bancos de dados não relacionais, conhecidos como *NoSQL*.

Apesar do nome, o termo *NoSQL* pode levar facilmente a um mal entendimento do seu significado. Mesmo que, inicialmente, a visão de um sistema que não possui um modelo relacional e não necessitava de *SQL* fosse predominante entre os criadores de sistemas não relacionais (HURWITZ et al., 2013), o *NoSQL* também é

conhecido como “*Not Only SQL*” já que o *SQL* não é totalmente evitado. Enquanto existem bancos de dados totalmente não relacionais, alguns simplesmente evitam funcionalidades específicas de sistemas relacionais, como esquemas de tabelas fixas e operações de união (CHEN; ZHANG, 2014).

Sistemas relacionais e não relacionais possuem fundamentos parecidos e a diferença entre eles se encontra na maneira como essas características fundamentais são implementadas. Sistemas não relacionais possuem como característica principal a escalabilidade, capacidade de manipular dados em de múltiplos bancos simultaneamente sem considerar as limitações físicas da infraestrutura subjacente. Outra característica importante é a consistência eventual, que é responsável pela resolução de conflitos quando os dados são movimentados em uma implementação distribuída (HURWITZ et al., 2013).

Existem muitos modelos de implementação de bancos de dados não relacionais, como os modelos baseados em colunas ou em grafos. O modelo chave-valor é um dos modelos mais implementados em bancos de dados *NoSQL*. Nele, os dados são tratados como um *array* associativo. Nesse modelo, segundo Hurwitz et. al. (2013), não existe a necessidade de um esquema, mas ele oferece grande flexibilidade e escalabilidade.

Warden (2011) cita que é impossível executar consultas em bancos de dados baseados unicamente no modelo chave-valor e que o código da aplicação deve ser capaz de manipular a construção de qualquer operação complexa que não sejam chamadas primitivas. Porém, isso pode ser visto como uma vantagem. Em um modelo mais simples as características de performance são muito previsíveis e é possível criar as operações mais complexas usando a mesma linguagem de programação da aplicação.

Outro modelo bastante utilizado em bancos de dados não relacionais é o modelo orientado a documentos que, ao contrário do modelo relacional, insere a informação no formato de uma linha de valores em uma tabela previamente definida pelo usuário, o modelo orientado a documentos não é tão rígido. Teoricamente, cada dado pode conter um conjunto de valores nomeados completamente diferente de outros dados, apesar de que, na prática, a camada de aplicação possua um esquema informal (WARDEN, 2011).

Entre as vantagens do modelo orientado a documentos, Warden (2011) cita a flexibilidade, pois a adição ou remoção ao equivalente de colunas pode ser efetuada

sem penalidades, desde que a camada de aplicação não necessite dos valores que foram removidos.

3.3 ANÁLISE EM BIG DATA

Assim como as ferramentas tradicionais de armazenamento e processamento, as ferramentas de análise de dados também não estão preparadas para manipular e extrair valor da quantidade cada vez maior de dados gerados pelos mais diversos sistemas, dispositivos e usuários. O *Big Data* também trouxe novos desafios para a área de análise de dados. Chandarana e Vijayalakshmi (2014) citam como desafios da análise em *Big Data* a privacidade e segurança, o gerenciamento e compartilhamento dos dados, o crescimento e expansão dos dados, velocidade e escala.

Além disso, a análise pode ser feita tanto em dados estruturados quanto em dados não estruturados. O tipo de análise que precisa ser feita depende altamente dos resultados a serem obtidos, ou seja, da tomada de decisão. Isso pode ser feito incorporando volumes de dados massivos na análise ou determinando previamente quando *Big Data* é relevante.

No entanto, as oportunidades provenientes da análise em *Big Data* são enormes. Hurwitz et. al. (2013) afirmam que a capacidade de analisar *Big Data* permite a expansão do tipo de análise que pode ser feita. Não existe mais a limitação de utilizar amostras de grandes conjuntos de dados, agora é possível analisar dados mais detalhados e completos.

3.4 BIG DATA E REDES SOCIAIS

O *Big Data* em redes sociais vem, segundo Zhao, Heuvel e Ye (2013), atraindo a atenção não só de pesquisadores da área de tecnologia da informação, mas também das áreas de economia, sociologia, psicologia, entre outras. O fenômeno conhecido como *Social Media Big Data* vem sendo crucial na revelação de detalhes do

comportamento humano, possibilitando a observação do fenômeno social em um nível nunca antes imaginado (TUFEKCI, 2014).

A pesquisa de redes sociais evolui para um problema de *Big Data* quando pesquisadores e empresas esperam prever um comportamento para alcançar uma melhora em *marketing*, vendas e comércio eletrônico (TAN et al., 2013). Como muitas redes sociais possuem dezenas ou até centenas de milhões de usuários, as amostras dos dados são cruciais para muitos estudos. Como muitas plataformas de redes sociais geram vastas quantidades de dados, muitos pesquisadores focam em subconjuntos de dados relacionados a eventos ou atividades específicas. Isso acaba diminuindo o tamanho do conjunto de dados a ser analisado (GAO; QIU, 2014).

Apesar do grande interesse que o *Social Media Big Data* vem atraindo, a pesquisa nessa área está longe de ser completa. Zhao, Heuvel e Ye (2013) citam o problema de pesquisas anteriores focadas no impacto da transmissão social em vez de suas causas. Comportamentos são gravados implicita e dispersamente em dados transacionais durante operações de negócios.

4 FERRAMENTAS PARA BIG DATA

Essa seção apresenta as principais ferramentas utilizadas em *Big Data*, seus componentes, aplicações, vantagens e como elas trabalham em conjunto para criarem soluções computacionais para gerenciamento e processamento de dados em *Big Data*.

4.1 APACHE HADOOP

Os *data warehouses* empresariais existentes se destacam no processamento de dados estruturados e podem armazenar vastas quantidades de dados. Mas essas vantagens tem um custo. A necessidade por estrutura restringe os tipos de dados que podem ser processados e impõe uma inércia que faz com que *data warehouses* não sejam recomendados para exploração ágil de grandes quantidades de dados heterogêneos. É nessa situação que o *Hadoop* pode fazer uma grande diferença (DUMBILL et al., 2012).

O *Apache Hadoop* é um projeto *open-source* utilizado no desenvolvimento de software para computação distribuída de forma escalável e confiável. Ele permite o processamento distribuído de grandes conjuntos de dados através de *clusters*, utilizando um modelo de programação simples (PATEL; BIRLA; NAIR, 2012).

Além disso, o *Hadoop* ajuda na solução dos três grandes desafios criados pelo *Big Data* (SINGH; KAUR, 2014): escalabilidade horizontal; velocidade de acesso; variedade dos dados. A escalabilidade horizontal de grandes conjuntos de dados arbitrários alcançada com o *Hadoop* ajuda a solucionar o problema do volume dos dados. O problema da velocidade dos dados é solucionado com a capacidade do *Hadoop* em gerenciar níveis elevados de dados vindos de sistemas de grandes proporções. Por fim, a variedade é solucionada já que o *Hadoop* suporta tarefas complexas para manipular dados não estruturados.

Mas para tirar proveito das potencialidades oferecidas pelo *Apache Hadoop*, as companhias precisam entender os dados que estão sendo coletados e como utilizá-los no contexto do modelo de negócios (HURWITZ et al., 2013).

O *Apache Hadoop* utiliza duas ferramentas fundamentais para conseguir uma computação distribuída de forma confiável e escalável: o *Hadoop Distributed File System* (HDFS) e o motor *MapReduce*. O HDFS é um sistema de arquivos distribuído utilizado para o armazenamento de dados e o *MapReduce* é um algoritmo de processamento de dados distribuído.

4.2 APACHE MAHOUT

O *Mahout* é uma biblioteca de aprendizado de máquina, *open source*, gerenciada pela Fundação *Apache*. Inicialmente, o *Mahout* era um subprojeto do *Apache Lucene*, um motor de busca que fornece implementações avançadas de busca e mineração de texto. No universo da ciência da computação, esses conceitos são relacionados a técnicas de aprendizado de máquina. Isso resultou na criação de um subprojeto, já que muitos colaboradores do *Apache Lucene* demonstraram mais interesse nas áreas do aprendizado de máquina (OWEN et al., 2012).

A biblioteca *Apache Mahout* implementa, basicamente, algoritmos de *clustering*, classificação e recomendação. Como o *Mahout* executa seus algoritmos em grandes conjuntos de dados, as soluções de aprendizado de máquina, desenvolvidas com o auxílio da biblioteca, devem ser escaláveis. Para alcançar essa escalabilidade, grande parte do código é escrito na forma de *jobs* paralelizáveis pelo *Hadoop* (WARDEN, 2011).

4.2.1 Recomendação de Conteúdo

Considerados um dos três pilares das implementações de aprendizado de máquina do *Apache Hadoop*, filtragem colaborativa e recomendação são técnicas usadas para alcançar um melhor entendimento dos gostos de uma pessoa, além de encontrar novos conteúdos para usuários, de forma automática (OWEN et al., 2012).

A filtragem colaborativa produz recomendações baseadas apenas no conhecimento das relações dos usuários com itens. Essa técnica não requer

conhecimento das propriedades dos itens, o que é uma vantagem, já que não requer que informações sobre os atributos façam parte da entrada (OWEN et al., 2012).

Ao usar a similaridade entre dados como a preferência de usuários e itens, os algoritmos de filtragem colaborativa são capazes de recomendar itens de maneira mais efetiva. As duas categorias mais amplas de algoritmos de filtragem colaborativa são os baseados no usuário e baseados no item (OWEN et al., 2012).

4.2.2 Recomendação Baseada no Usuário

A principal característica de um algoritmo de recomendação baseado no usuário é que ele é baseado na noção de que existem similaridades entre usuários (OWEN et al., 2012). Para realizar predições do que cada usuário irá gostar no futuro, um algoritmo *user-based* identifica correlações entre diferentes usuários com base em preferências do passado que são similares (CASINELLI, 2014).

Uma vizinhança é formada por um grupo de usuários com preferências similares, e normalmente é construída por um sistema de recomendação para ajudar a recomendar itens. A vantagem na utilização de recomendação baseada no usuário é que a recomendação será especifica para cada usuário e se adaptará ao usuário conforme novas recomendações forem introduzidas no sistema de recomendação (CASINELLI, 2014). A Figura 2 ilustra um exemplo de recomendador utilizando uma vizinhança.

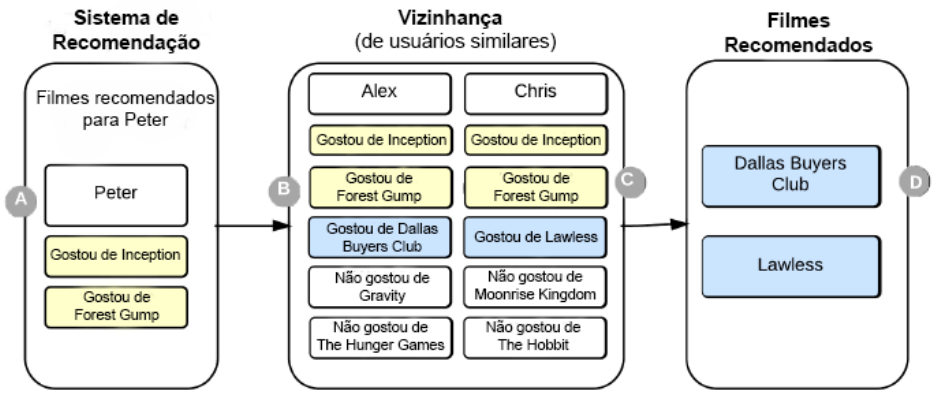

Figura 2 - Recomendação Baseada no Usuário

Fonte: adaptado de Casinelli (2014)

A implementação, do algoritmo de recomendação baseada no usuário, feita pelo *Mahout*, considera cada item conhecido que o usuário ainda não expressou preferência como um candidato para recomendação. Depois disso, é visto o valor de preferência do item candidato de todos os usuários que expressaram preferência por esse item. Uma média ponderada é criada utilizando todos os valores encontrados, na qual o peso de um valor de preferência é baseado na similaridade do usuário em relação ao usuário alvo (OWEN et al., 2012).

Examinar todos os itens seria um processo muito lento, por isso uma vizinhança de usuários similares é computada primeiro e somente os itens conhecidos por esses usuários são considerados. Usuários similares são encontrados primeiro e somente após isso é feito a busca de itens que esses usuários possuem interesse. Esses itens se tornam candidatos à recomendação (OWEN et al., 2012).

4.2.3 Recomendação Baseada no Item

A recomendação baseada no item é derivada da similaridade entre itens, ao invés da similaridade entre usuário. A diferença básica entre *user-based* e *item-based* é que a recomendação baseada no usuário encontra usuários similares e analisa quais são suas preferências. Já a recomendação baseada no item visualiza as preferências do usuário e então busca itens similares (OWEN et al., 2012).

Como a similaridade entre itens é usada na construção de recomendações, ao invés da construção de uma vizinhança são feitas correlações entre preferências de itens. A vantagem na utilização de algoritmos *item-based* está na escala menor dos itens, já que os itens tendem a crescer em um ritmo mais lento do que os usuários (CASINELLI, 2014). O funcionamento de um sistema de recomendação baseado no item é ilustrado na Figura 3.

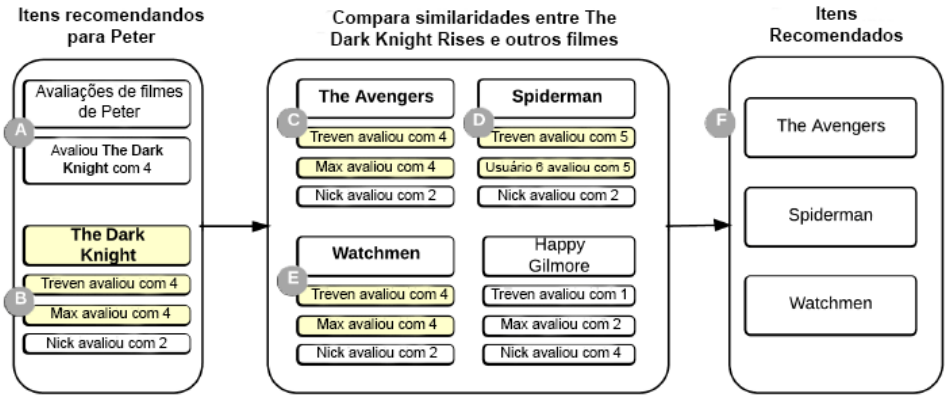

Figura 3 - Recomendação Baseada no Item
Fonte: adaptado de Casinelli (2014)

A implementação do algoritmo pelo *Mahout* é feita de maneira similar a implementação do algoritmo *user-based*. A diferença é que o *item-based* foca na similaridade entre preferências de diferentes itens e não na similaridade entre preferências de diferentes usuários. Ao criar uma recomendação baseada no item, o *Mahout* utiliza um processo de estimativa ao aplicar o código em todos os outros itens do modelo de dados (CASINELLI, 2014).

4.2.4 Medidas de Similaridade

A implementação de funções para o cálculo de similaridade é uma das partes mais importantes. Tanto a recomendação *user-based* quanto a *item-based* dependem muito desse componente. Sem uma noção confiável e efetiva de quais usuários ou itens são similares, essas abordagens são falhas (OWEN et al.,2012). O *Mahout* possui implementações de vários algoritmos de similaridade amplamente utilizados e permite que desenvolvedores os conectem em sistemas de recomendação, de modo a identificar vizinhanças similares para usuários ou calcular similaridade entre itens (CASINELLI, 2014).

Entre os algoritmos de similaridade implementados pelo *Mahout* podem ser citados:

- *Pearson Correlation* - é um número entre -1 e 1, que mede a tendência de duas séries de números de se moverem juntas, proporcionalmente, de tal

forma que exista uma relação aproximadamente linear entre os valores das duas séries. Esse conceito, amplamente utilizado em estatística, pode ser utilizado para medir a similaridade entre usuários (OWEN et al., 2012).

$$PC(w,u) = \frac{\sum_i \ (r_{w,i} - \bar{r}_w)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_i (r_{w,i} - \bar{r}_w)^2 \sum_i (r_{u,i} - \bar{r}_u)^2}} \quad \textbf{(1)}$$

Onde w e u representam os dois usuários ou itens para qual o coeficiente está sendo calculado, i é um item, $r_{w,i}$ e $r_{u,i}$ são avaliações de w e u para i, e $\bar{r}_w$ e $\bar{r}_u$ são classificações médias para usuário (ou item) w e u, respectivamente (CASINELLI, 2014).

A implementação da similaridade *Pearson Correlation*, ilustrada na Equação (1), fornece uma extensão da computação padrão, chamada de *weighting*, que mitiga o problema que essa medida de similaridade possui, o de não refletir diretamente, o número de itens sobre o qual ela é calculada (OWEN et al., 2012).

- *Euclidean Distance* – essa implementação é baseada na distância entre usuários. Imaginando os usuários como pontos em um espaço de *n* dimensões, onde *n* é o número de itens, com os valores de preferência como coordenadas. Essa medida de similaridade computa a distância euclidiana *d* entre dois "pontos usuários". A forma como a medida é calculada é mostrada na Equação (2).

$$D = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2} \quad \textbf{(2)}$$

Onde n é o número de dimensões (atributos) e A_i e B_i são i-ésimos atributos (componentes) dos dados A e B.

- *Spearman Correlation* – uma das variações da medida *Pearson Correlation*. A única diferença é que as avaliações dos itens são recalculadas de acordo com *ranking* das avaliações iniciais antes da expansão do cálculo da correlação (GUO, 2014).
- *Tanimoto Coefficient* – um dos tipos de implementação que ignora valores de preferência completamente, importando apenas se o usuário expressa uma preferência. Essa medida é a razão do tamanho da intersecção em

relação ao tamanho da união dos itens que dois usuários demonstraram preferência (Owen et al., 2012). A Equação (3) mostra como essa medida de similaridade é calculada.

$$T(X,Y) = \frac{X \cap Y}{X \cup Y} \quad \textbf{(3)}$$

Onde X e Y definem elementos nos conjuntos de dados e o coeficiente é definido como o tamanho da intersecção dividido pelo tamanho da união dos conjuntos de dados.

- *Log Likelihood* – medida de similaridade que, assim como a medida *Tanimoto Coefficient*, se baseia no número de itens em comum entre dois usuários, mas seu valor é mais uma expressão do quão improvável dois usuários tenham sobreposição, dado o número total de itens e o número de itens que cada usuário possui preferência (OWEN et al., 2012).

4.2.5 Avaliação de Recomendadores

Uma forma de avaliar um recomendador é calculando a qualidade dos valores de preferência estimados, ou seja, avaliando o quão perto as preferências estimadas estão das preferências reais (OWEN et al., 2012).

Para fazer esse cálculo, o *Mahout* implementa o `RecommenderEvaluator`, que utiliza a diferença média absoluta e a *Root-Mean-Square* para calcular a qualidade das recomendações. O método `evaluate()` realiza esse cálculo dividindo o conjunto de dados em treino e teste, criando as recomendações para o conjunto de teste e comparando os valores estimados com os dados de teste (OWEN et al., 2012).

O *score* é o resultado dessa avaliação, indicando o nível de desempenho do recomendador. O que esse valor significa depende da implementação utilizada. Por exemplo, com a diferença média absoluta, o valor do *score* indica o quanto a preferência estimada é diferente da preferência real (OWEN et al., 2012). Quanto menor o *score*, melhor.

Métricas de recuperação de informação clássicas, como precisão e cobertura, também podem ser utilizadas na avaliação de recomendadores. No contexto de recomendação, precisão é a proporção de recomendações que estão no topo da lista de recomendações e que são boas recomendações. Cobertura é a proporção de boas recomendações que aparecem no topo da lista de recomendações (OWEN et al., 2012). A medida F1 é definida como a média harmônica da precisão e cobertura (SASAKI, 2007).

Para calcular precisão e cobertura o *Mahout* utiliza o método `evaluate()` do `RecommenderIRStatsEvaluator` que escolhe um limiar, por usuário, para definir o que é uma boa recomendação. Esse limiar é igual ao valor de preferência média do usuário somado a um desvio padrão (OWEN et al., 2012).

5 MATERIAL E MÉTODOS

Neste capítulo são descritas as ferramentas utilizadas na construção do componente de recomendação, suas características, e a maneira como foram utilizadas para alcançar um resultado satisfatório no desenvolvimento do componente.

5.1 ARQUITETURA DO SISTEMA

As próximas seções apresentam os artefatos que descrevem a arquitetura do sistema e seu comportamento. A *Unified Modeling Language*[3] foi utilizada na modelagem dos diagramas citados a seguir, construídos com o auxílio da ferramenta *Astah*[4].

Foram elaborados os requisitos da aplicação para identificar as necessidades que serão atendidas pelo componente a ser desenvolvido. Os requisitos da aplicação são:

- Extração dos dados – os dados utilizados para criar as recomendações foram extraídos de redes sociais;
- Mineração de sentimentos – foi implementada uma solução que calculeou a preferência de um usuário por um item;
- Utilização de vizinhanças – Foram implementados dois tipos de vizinhança para os recomendadores que utilizam vizinhança para realizar recomendações;
- Avaliação do recomendador – Foi implementado um método que avaliou a qualidade das recomendações, além disso, métricas de recuperação de informações clássicas, como precisão cobertura e medida F1 também foram utilizadas;

O diagrama de casos de uso, mostrado na Figura 4, foi elaborado com o propósito de descrever o que o componente deve fazer.

[3] http://www.uml.org/
[4] http://astah.net/

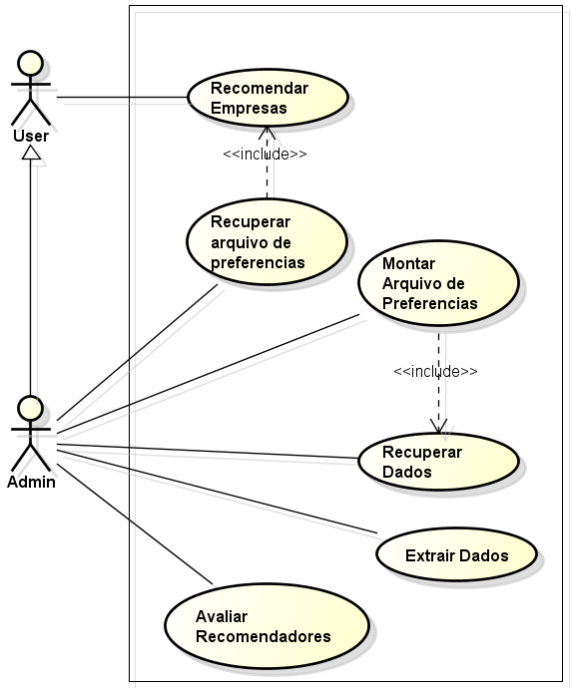

Figura 4 - Diagrama de casos de uso
Fonte: Autoria Própria

Foram identificados dois atores, o usuário e o administrador do sistema. O usuário faz parte apenas do caso de uso “Recomendar Empresas”, enquanto o administrador é responsável por alimentar o recomendador com dados de redes sociais, processo que é feito através de todos os outros casos de uso, melhorando assim as recomendações que serão feitas para o usuário.

5.1.1 Diagrama de Classe

O diagrama de classes foi elaborado para oferecer uma visão geral do componente desenvolvido, descrevendo suas diferentes partes e como elas se relacionam. As Figuras 5,6 e 7 mostram a visão detalhada de cada um dos pacotes que formam o diagrama e a visão geral do diagrama de classes pode ser vista no Apêndice A.

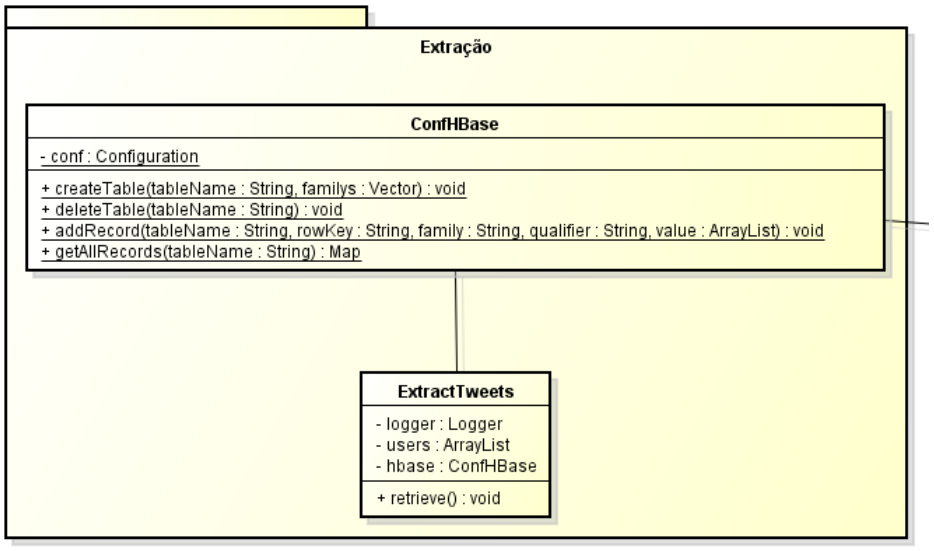

Figura 5 - Diagrama de classes: Pacote Extração

Fonte: Autoria Própria

A Figura 5 ilustra o pacote Extração, que contém as classes utilizadas para realizar a extração e armazenamento dos dados do *Twitter*. A classe `ExtractTweets` possui as configurações necessárias para a biblioteca *Twitter4J* faça a extração dos dados e armazene os dados, utilizando a classe `ConfHBase` para fazer a comunicação com o banco de dados *HBase*.

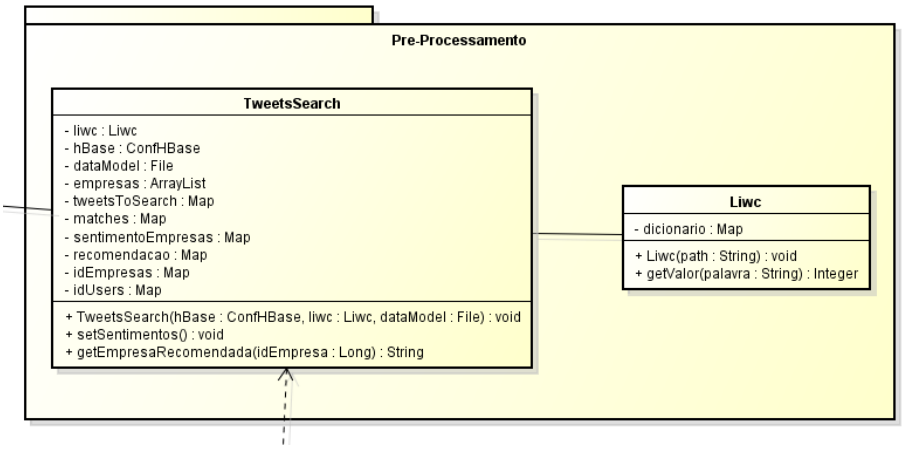

Figura 6 - Diagrama de classes: Pacote Pré-processamento

Fonte: Autoria Própria

A Figura 6 mostra o pacote Pré-processamento, com as classes utilizadas para fazer o pré-processamento dos dados salvos no banco de dados. A classe `Liwc` é responsável por calcular a polaridade de todos os *tweets* salvos no processo de

extração e armazenamento dos dados. A classe `TweetsSearch` utiliza a classe `Liwc` e calcula a preferência de cada usuário por cada empresa no arquivo de empresas. O arquivo de preferências é gerado ao final da execução dessa classe.

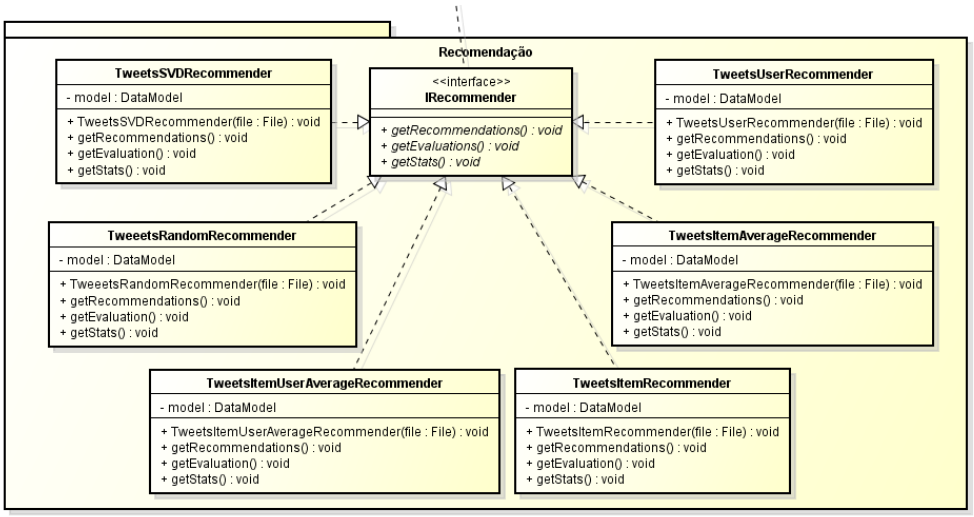

Figura 7 - Diagrama de classes: Pacote Recomendação
Fonte: Autoria Própria

Por fim, a Figura 7 ilustra o processo de recomendação. A interface `IRecommender` foi criada e seis tipos de recomendações foram implementadas. Todas as implementações dessa interface recebem o arquivo de preferências como entrada e a saída são as recomendações e as avaliações de *score*, precisão, cobertura e medida F1.

5.2 IMPLEMENTAÇÃO

Nessa seção são descritos todos os passos da construção do sistema de recomendação, bem como as ferramentas utilizadas na implementação do mesmo. A descrição do sistema é mostrada na Figura 8.

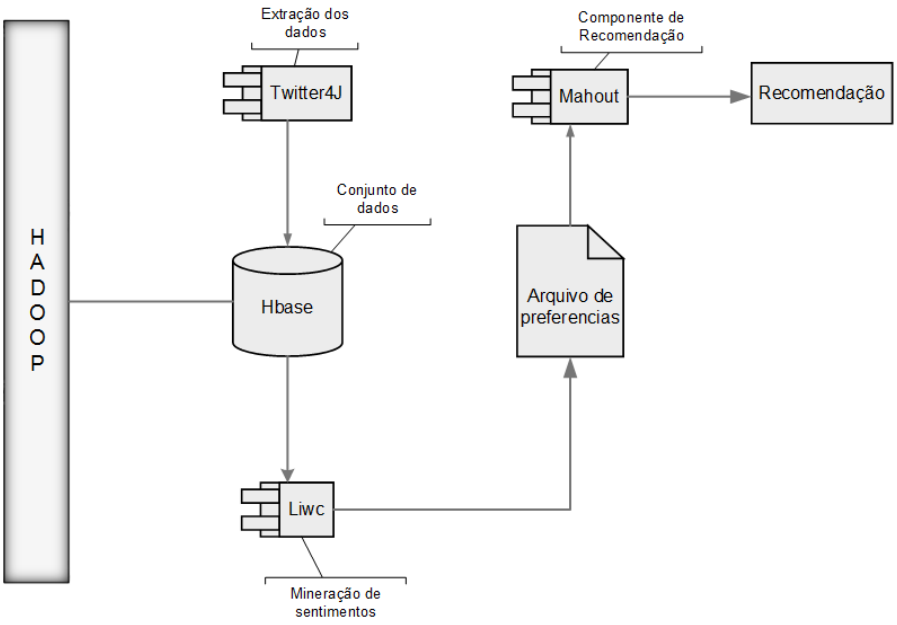

Figura 8 - Visão geral do Sistema

Fonte: Autoria Própria

5.2.1 Extração e armazenamento dos dados

O *Twitter4J*[5] é uma biblioteca JAVA utilizada na integração da API (*Application Programing Interface*) oficial do *Twitter* com aplicações e foi utilizada na extração dos dados. As *queries*[6] de buscas utilizadas para montar o banco de dados envolviam empresas de tecnologia como *Samsung, Google, Microsoft* e *Motorola*. As buscas foram feitas para 614 empresas, salvas em um arquivo de texto criado previamente.

[5] http://twitter4j.org/en/index.html

[6] Uma query, nesse contexto, é a palavra usada para fazer a busca. O *Twitter4J* vai buscar pelos *tweets* mais recentes que tenham a query em seu texto e retorná-los para o usuário.

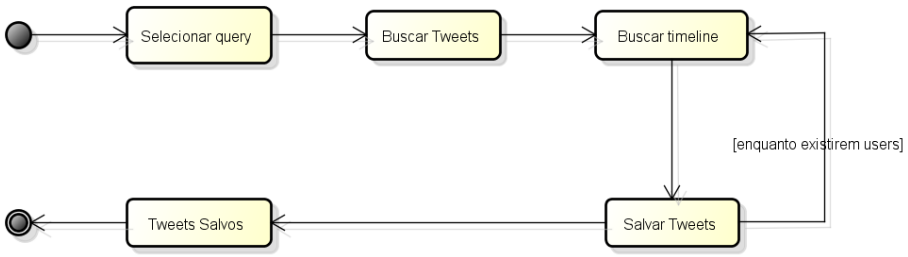

Figura 9 - Extração e armazenamento dos dados
Fonte: Autoria Própria

Conforme o diagrama mostrado na Figura 9, primeiramente foi feita uma busca pelos quinze primeiros *tweets* que continham em seu texto a *query* de busca. Os usuários que postaram esses quinze *tweets* são salvos em uma lista de *Users*, uma implementação do *Twitter4J* que salva todas as informações relevantes de um usuário, como seu id e *screenName*, que é o "nome" único do usuário.

Após isso, é construída uma lista com os primeiros dois mil *tweets* da *timeline* de cada um dos usuários na lista de *Users*. Uma *timeline* é a coleção de todos os *tweets* postados por um usuário, ordenados pela data de postagem. O usuário, referenciado pelo *screenName*, e sua respectiva lista de *tweets* são então salvos no banco de dados.

Os dados extraídos foram armazenados utilizando o *HBase*, um banco de dados distribuído, não relacional, que utiliza o HDFS para armazenamento com persistência, e que fornece acesso aos dados de forma aleatória e em tempo real tanto para escrita quanto para leitura dos dados (HURWITZ et al., 2013). O *Hadoop* foi instalado de forma pseudo-distribuída no sistema, já que o *HBase* utiliza o HDFS para armazenar os dados.

A tabela criada para salvar os dados extraídos do *Twitter* recebeu o nome de '*Tweets*'. Essa tabela possui três atributos: `row key`, `column family` e `value`. A `row key` é a chave única que identifica o armazenamento lógico de todas as colunas em uma linha.O HBase armazena os dados organizados pela `row key`, garantindo velocidade no acesso de leitura (KERZNER; MANIYAM, 2014).

Nessa tabela, a `row key` é o *screenName* do usuário. A `column family` da tabela recebeu o nome de '*Text*'. No *HBase*, linhas são compostas por colunas, e essas, por sua vez, são agrupadas em *column families*. Isso ajuda a construir fronteiras semânticas entre os dados, aplicar compressão ou denotar que os dados

fiquem em memória (GEORGE, 2011). Por último, o `value` de cada `row key` na tabela é a lista de *tweets* extraídos da *timeline* de cada usuário.

5.2.2 Mineração de sentimentos e pré-processamento dos dados

Para verificar a preferência de um usuário por uma empresa, foram feitos os seguintes passos:

- Recuperação dos dados;
- Todos os *tweets* de cada usuário foram comparados com a lista de 614 empresas;
- Se o nome de uma empresa estiver no texto de um *tweet*, ele é salvo em uma lista contendo todos os outros *tweets* que fazem menção a essa empresa.
- O processo do passo anterior é repetido até que exista uma lista para cada empresa, contendo todos os *tweets* com o nome dessa empresa em seu texto.

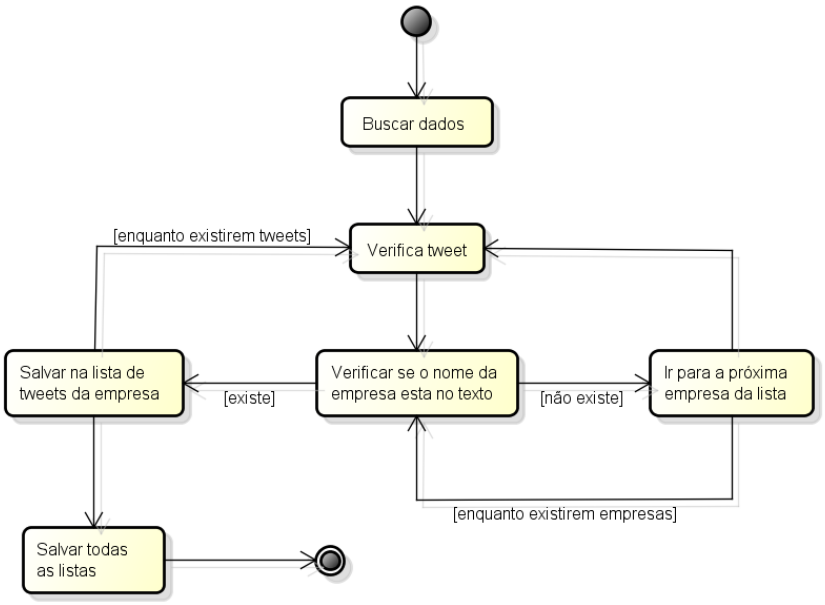

Figura 10 - Pré-processamento 1

Fonte: Autoria Própria

Esse processo, ilustrado na Figura 10, descreve apenas os primeiros passos na mineração de sentimentos e pré-processamento dos dados. A continuação envolve os seguintes passos, ilustrados na Figura 11:

- Todas as listas de cada usuário são passadas para o *Liwc* calcular a preferência de cada usuário para todas as empresas que ele citou em algum de seus *tweets*;
- A polaridade de cada palavra é determinada pelo *Liwc*. A soma de todas as polaridades é feita e então dividida pela quantidade de palavras no texto. A média da polaridade é então adicionada à soma final dessa lista. Esse processo é repetido em todos os *tweets* da lista. Quando uma palavra com polaridade é encontrada, é verificado se a palavra não aparece anteriormente. Em caso positivo, a polaridade da palavra é invertida;
- Por fim, a soma final é dividida pelo número de *tweets* na lista, definindo assim a preferência do usuário por cada empresa que ele tenha citado em seus *tweets*.

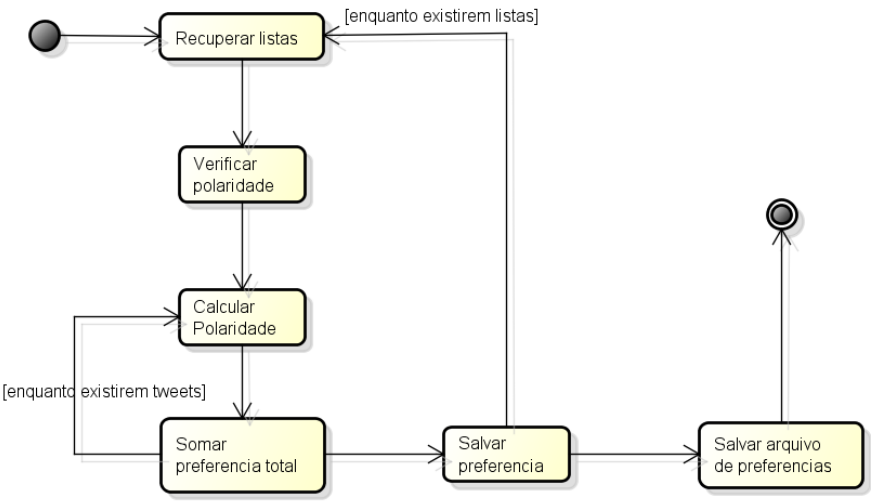

Figura 11 - Pré-processamento 2
Fonte: Autoria Própria

O *Liwc* é um léxico que atribui um valor de acordo com a polaridade da palavra: 1 para palavras com polaridade positiva, 0 para palavras com polaridade neutra e -1 para palavras com polaridade negativa.

Como o *Mahout* faz as recomendações usando valores numéricos, foi preciso definir uma identificação numérica única para cada usuário e para cada empresa. Após isso, foi montado um arquivo de texto contendo o *id* do usuário, o *id* da empresa e a preferência, calculada pelo *Liwc,* do usuário por essa empresa. O formato desse arquivo é mostrado na Figura 12.

#ID Usuário	ID Empresa	Preferência
0	632	0.026
0	572	0.105
0	299	-0.071
0	219	0.0
1	243	0.058
1	592	0.0
1	452	0.037
2	144	0.114
2	190	0.107
2	311	-0.020

Figura 12 - Modelo do arquivo de preferências
Fonte: Autoria Própria

5.2.3 Recomendação

A recomendação dos dados foi feita utilizando a biblioteca *Apache Mahout*. Foram implementados seis tipos de recomendadores: `ItemAverage`, `ItemBased`, `ItemUserAverage`, `Random`, `SVD` e `UserBased`. Todos os recomendadores utilizam o arquivo de preferências, mostrado na Figura 11, para fazer recomendações.

Nos experimentos com os recomendadores que utilizam vizinhança e medida de similaridade para realizar recomendações, `UserBased` e `ItemBased`, foram usadas as medidas *Pearson Correlation, Pearson Correlation* com *weighting, Euclidean Distance, Euclidean Distance* com *weighting, Tanimoto Coefficient, Log Likelihood, Spearman Correlation*. As vizinhanças *NearestNUser*, com uma vizinhança de tamanho 5, e *ThresholdUser*, com um *threshold* de 0.5, foram utilizadas nos experimentos.

O avaliador foi construído utilizando a diferença média absoluta e a divisão do conjunto de dados entre treino e teste foi de 66% e 33%, valores padrões na

bibliografia. Outras medidas de avaliação utilizadas foram precisão, cobertura e medida F1.

6 RESULTADOS OBTIDOS

Com o auxílio da biblioteca *Twitter4J* foram extraídos 1,239,515 *tweets*. Utilizando o léxico *Liwc* foram identificadas 12372 preferências dos usuários por alguma das 614 empresas de uma lista montada previamente. As preferências dos usuários foram salvas em um arquivo de texto, juntamente com o ID do usuário e o ID da empresa. Esse arquivo de preferências segue o modelo apresentado na Figura 16.

Foram criados seis tipos de recomendadores que utilizam o arquivo de texto com as preferências para realizar recomendações. Cada recomendador utiliza diferentes tipos de medidas de similaridade e vizinhanças para fazer uma recomendação. Os resultados para cada tipo de recomendador são discutidos a seguir.

6.1 USERBASEDRECOMMENDER

Nos experimentos com o recomendador do tipo `UserBasedRecommender` foram utilizadas sete medidas de similaridade e dois tipos de vizinhança. Os resultados são mostrados nas Tabelas 1 e 2.

Tabela 1 - Resultado da avaliação do recomendador UserBased utilizando vizinhança do tipo NearestNUser

Medida de Similaridade	Score	Precisão	Cobertura	Medida F1
PearsonCorrelation	0.0421698	0.1276223	0.1489698	0.1374723
PearsonCorrelation com Weighting	0.0406847	0.1276223	0.1489698	0.1374723
EuclideanDistance	0.0330997	0.1503690	0.1497622	0.1500650
EuclideanDistance com Weighting	0.0303876	0.1503690	0.1497622	0.1500650
TanimotoCoefficient	**0.0207154**	**0.2465277**	**0.2591125**	**0.2526635**
LogLikelihood	0.0224410	0.1921221	0.2218700	0.2059273
SpearmanCorrelation	0.0420553	0.1296900	0.1671949	0.1460735

Fonte: Autoria Própria

Tabela 2 - Resultado da avaliação do recomendador UserBased utilizando vizinhança do tipo ThresholdUser

Medida de Similaridade	Score	Precision	Recall	F1 Measure
PearsonCorrelation	0.0339027	0.0471380	0.0522979	0.0495841
PearsonCorrelation com Weighting	0.0323937	0.0462962	0.0515055	0.0487621
EuclideanDistance	0.0349688	0.0103011	0.0103011	0.0103011
EuclideanDistance com Weighting	0.0354754	0.0103011	0.0103011	0.0103011
TanimotoCoefficient	**0.0519710**	**0.2849056**	**0.1410459**	**0.1886824**
LogLikelihood	0.0335943	0.0103011	0.0103011	0.0103011
SpearmanCorrelation	0.0336996	0.0467511	0.0538827	0.0500642

Fonte: Autoria Própria

Analisando os resultados das Tabelas 1 e 2 é possível afirmar que, para o recomendador `UserBasedRecommender`, a melhor medida de similaridade a ser utilizada é a *Tanimoto Coefficient* e o melhor tipo de vizinhança é a *NearestNUser*, já que ambas obtiveram os menores resultados de score e os maiores resultados em precisão, cobertura e medida F1. Apesar da vizinhança do tipo *ThresholdUser* gerar valores maiores de precisão, os resultados de *score*, cobertura e medida F1 são inferiores aos resultados alcançados quando a vizinhança do tipo *NearestNUser* é utilizada. Isso justifica a escolha do *NearestNUser* como o melhor tipo de vizinhança para o `UserBasedRecommender`.

6.2 ITEMBASEDRECOMMENDER

Como o `ItemBasedRecommender` não utiliza uma vizinhança para fazer recomendações, os experimentos com esse tipo de recomendador utilizaram apenas as seis medidas de similaridade. Os resultados, ilustrados na Tabela 3, indicam que a melhor medida de similaridade para esse tipo de recomendador é a *Pearson Correlation* com *weighting*. Apesar dessa medida não ter o menor resultado no *score*, os resultados de precisão, cobertura e medida F1 são muito melhores quando comparados aos de outras medidas, justificando assim essa escolha.

Tabela 3 - Resultado da avaliação do recomendador ItemBased

Medida de Similaridade	Score	Precisão	Cobertura	Medida F1
Pearson Correlation	0.1046977	0.0031695	0.0047543	0.0038034
Pearson Correlation com Weighting	**0.1049666**	**0.0047543**	**0.0063391**	**0.0054335**
Euclidean Distance	0.0382502	0.0000790	0.0000790	0.0000790
Euclidean Distance com Weighting	0.0375526	0.0000790	0.0000790	0.0000790
Tanimoto Coefficient	0.0384247	0	0	*
Log Likelihood	0.0389875	0.0000790	0.0000790	0.0000790

*** Nesse caso o recomendador não foi capaz de calcular um número válido**

Fonte: Autoria Própria

6.3 OUTROS RECOMENDADORES

O *Mahout* possui várias implementações de recomendadores que não utilizam medida de similaridade e vizinhança para realizar recomendações. Os resultados desses recomendadores são mostrados na Tabela 4.

Tabela 4 - Resultado da avaliação dos recomendadores que não utilizam medida de similaridade e vizinhança

Recomendadores	Score	Precision	Recall	F1 Measure
ItemAverageRecommender	0.0347688	0	0	*
ItemUserAverageRecommender	0.0369699	0	0	*
SVDRecommender	**0.0344743**	**0.0483359**	**0.0673534**	**0.0562816**
RandomRecommender	0.1974622	0.0039619	0.0039619	0.0039619

*** Nesse caso o recomendador não foi capaz de calcular um número válido**

Fonte: Autoria Própria

Analisando os resultados da tabela 4, é possível afirmar que o `SVDRecommender` é a melhor escolha de recomendador que não utiliza medida de similaridade e vizinhança, já que este conseguiu os maiores valores de score, precisão, cobertura e medida F1.

6.4 RESULTADO DAS RECOMENDAÇÕES

Após a análise dos resultados de todos os experimentos foi possível concluir que, utilizando o recomendador `UserBasedRecommender`, a medida de similaridade *Tanimoto Coefficient* e a vizinhança do tipo *NearestNUser*, para usuários que demonstraram interesse na *Google*, foram recomendadas as seguintes empresas:

- *Sony;*
- *Apple;*
- *Nintendo*.

Para usuários que demonstraram interesse na *Motorola*, foram recomendadas as empresas:

- *Samsumg*;
- *Nokia*

As recomendações para usuários que demonstraram interesse no *Samsumg*, foram:

- *Motorola;*
- *Sony*.

7 CONCLUSÃO

Este trabalho apresentou o uso das técnicas de mineração de sentimentos e recomendação de conteúdo para extrair conhecimento de dados criados em redes sociais, em um contexto de *Big Data*.

De modo geral, os resultados obtidos neste trabalho foram satisfatórios. Os valores de *score* foram bons na maioria dos recomendadores implementados, já que quanto menor o valor do *score* melhor, pois isso significa que as estimativas diferiram pouco dos valores de preferências (OWEN, et al., 2012). Porem os valores das outras medidas de avaliação – precisão, cobertura e medida F1 – ficaram abaixo do esperado. Refinar a solução de mineração de sentimentos pode ser uma das formas de melhorar os resultados do trabalho. Usar uma solução que demonstre da forma mais fiel possível qual a opinião de um usuário em relação a um item pode ter um grande impacto no resultado final. Outra forma pode ser utilizando um conjunto de dados maior. Extrair mais dados das redes sociais pode ajudar o recomendador a ter um entendimento maior sobre a preferência de um usuário em relação a um item, gerando recomendações melhores.

7.1 TRABALHOS FUTUROS

Para a continuação desse trabalho recomenda-se:

- Implementação do componente de recomendação, *Hadoop* e do banco de dados *HBase* em um cluster de computadores. Um ambiente distribuído é recomendado na solução de problemas de *Big Data*;
- Implementação de outras soluções de mineração de sentimentos;
- Extração de dados de outras redes sociais para visualizar, por exemplo: um usuário demonstrou interesse pela *Samsumg* no *Twitter* e foi recomendado para ele as marcas *Motorola* e *Sony*. Se um usuário do *Facebook* demonstrar um interesse parecido pela *Samsumg* as marcas *Motorola* e *Sony* também serão recomendadas para ele?

REFERÊNCIAS

ANDERSSON, E.; BOGREN, E.; BREDMAR, F. Scalable Machine Learning for Big Data, 2014.

AUE, A.; GAMON, M. **Customizing sentiment classifiers to new domains:** A case study. Proceedings of recent advances in natural language processing. [S.l.]: [s.n.]. 2005.

BALAGE, P. P.; PARDO, T. A. S.; ALUÍSIO, S. M. **An Evaluation of the Brazilian Portuguese LIWC Dictionary for Sentiment Analysis**. Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology. Fortaleza: [s.n.]. 2013. p. 2015-219.

CASINELLI, P. Evaluating and Implementing Recommender Systems As Web Services Using Apache Mahout, 2014.

CHANDARANA, P.; VIJAYALAKSHMI, M. **Big Data Analytics Frameworks**. 2014 International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA. [S.l.]: [s.n.]. 2014. p. 430-434.

CHEN, C. L. P.; ZHANG, C.-Y. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. **Information Sciences**, 21 January 2014. 314-247.

DONG, X. L.; SRIVASTAVA, D. **Big Data Integration**. ICDE Conference. [S.l.]: [s.n.]. 2013. p. 1245-1248.

DUMBILL, E. et al. **Big Data Now**. 1. ed. Sebastopol: O'Reilly Media,Inc, 2012.

ESTEVES, R. M.; RONG, C. **Using Mahout for clustering Wikipedia's latest articles**. Third IEEE International Conference on Cloud Computing Technology and Science. [S.l.]: [s.n.]. 2011. p. 565-569.

FACELI, K. et al. **Inteligência Artificial:** Uma Abordagem de Aprendizado de Máquina. Rio de Janeiro: LTC, 2011.

GAO, X.; QIU, J. **Supporting Queries and Analyses of Large-Scale Social Media Data with Customizable and Scalable Indexing Techniques over NoSQL Databases**. 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. [S.l.]: [s.n.]. 2014. p. 587-590.

GEORGE, L. **HBase:** The Definitive Guide. 1. ed. Sebastopol: O'Reilly Media, Inc., 2011.

GUO, S. **Analysis and evaluation of similarity metric in collaborative filtering recommender system**. [S.l.]: [s.n.], 2014.

HURWITZ, J. et al. **Big Data for Dummies**. Hoboken: John Wiley & Sons, Inc., 2013. 339 p.

HUTTO, C. J.; GILBERT, E. **VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text**. Georgia Institute of Technology. Atlanta, p. 10. 2014.

KATAL, A.; WAZID, M.; GOUDAR, R. H. Big Data: Issues, Challenges, Tools and Good Practices, 2013. 404-409.

KERZNER, M.; MANIYAM, S. **HBase Design Patterns**. Birmingham: Publishing Ltd., 2014.

LIU, Z.; YANG, P.; ZHANG, L. **A Sketch of Big Data Technologies**. 2013 Seventh International Conference on Internet Computing for Engineering and Science. Shanghai: Conference Publishing Services. 2013. p. 26-29.

MCAFEE, A.; BRYNJOLFSSON, E. Big Data: The Management Revolution. **Spotlight on Big Data**, October 2012. 61-68.

MITCHELL, T. M. **Machine Learning**. 1. ed. New York: McGraw-Hill, Inc., 1997.

OWEN, S. et al. **Mahout in Action**. Shelter Island: Manning Publication Co., 2012.

PALTOGLOU, G. et al. **Sentiment analysis of informal textual communication in cyberspace**. School of Computing and Information Technology, University of Wolverhampton. Wolverhampton, p. 15. 2010.

PATEL, A. B.; BIRLA, M.; NAIR, U. **Addressing Big Data Problem Using Hadoop and Map Reduce**. Nirma University International Conferenceon Engineering. [S.l.]: [s.n.]. 2012. p. 1-4.

PENNEBAKER, J. W.; BOOTH, R. J.; FRANCIS, M. E. **Linguistic inquiry and word:** Liwc. [S.l.]: [s.n.], 2001.

SASAKI, Y. **The truth of the F-measure**, Manchester, 26 October 2007. 1-5.

SINGH, K.; KAUR, R. **Hadoop:** Addressing Challenges of Big Data. IEEE International Advance Computing Conference. [S.l.]: [s.n.]. 2014. p. 686-689.

TAN, W. et al. Social-Network-Sourced Big Data Analytics. **IEEE Internet Computing**, September 2013. 62-69.

TAUSCZIK, Y. R.; PENNEBAKER, J. W. The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. **Journal of Language and Social Psychology**, p. 24-54, 2010.

TUFEKCI, Z. **Big Questions for Social Media Big Data:** Representativeness, Validity and Other Methodological Pitfalls. 8th International AAAI Conference on Weblogs and Social Media. [S.l.]: [s.n.]. 2014. p. 1-10.

VIEIRA, M. R. et al. **Banco de Dados NoSQL:** Concetos, Ferramentas, Linguagens e Estudos de Casos no Contexto de Big Data. Simpósio Brasileiro de Bando de Dados. [S.l.]: [s.n.]. 2012. p. 1-30.

WARDEN, P. **Big Data Glossary**. Sebastopol: O'Reilly Media, 2011.

ZAFARANI, R.; ABBASI, M. A.; LIU, H. **Social Media Mining:** An Introduction. [S.l.]: Cambridge University Press, 2014.

ZHANG, J.; HUANG, M. L. **5Ws Model for BigData Analysis and Visualization**. 2013 IEEE International Conference on Computacional Science and Engineering. [S.l.]: [s.n.]. 2013. p. 1021-1028.

ZHAO, Y. W.; HEUVEL, W.-J. V. D.; YE, X. **Exploring Big Data in Small Forms:** A Multi-layered Knowledge Extraction of Social Networks. 2013 IEEE International Conference on Big Data. [S.l.]: [s.n.]. 2013. p. 60-67.

APÊNDICE A – VISÃO GERAL DO DIAGRAMA DE CLASSES