

ERBD

XII ESCOLA REGIONAL DE BANCO DE DADOS

LONDRINA 2016



TEMA **DATA SCIENCE**

ANAIS



XII ESCOLA REGIONAL DE BANCO DE DADOS

13 a 15 de abril de 2016
Londrina – PR – Brazil

ANAIS

Promoção

Sociedade Brasileira de Computação – SBC
SBC Comissão Especial de Bancos de Dados

Organização

Universidade Estadual de Londrina - UEL
Universidade Estadual de Maringá - UEM
Universidade Tecnológica Federal do Paraná - UTFPR - Campus Cornélio Procópio

Comitê Diretivo da ERBD

Carmem Hara – UFPR (Presidente)
Fernando José Braz – IFC
Daniel Luis Notari – UCS

Chair Local

Daniel dos Santos Kaster

Comitê de Programa

Carmem Satie Hara (UFPR)

ISSN: 2177-4226

**Catalogação elaborada pela Divisão de Processos Técnicos da
Biblioteca Central da Universidade Estadual de Londrina**

Dados Internacionais de Catalogação-na-Publicação (CIP)

E74a Escola Regional de Banco de Dados (12. : 2016 : Londrina, PR)
Anais [da] XII Escola Regional de Banco de Dados [livro eletrônico] / promoção: Sociedade Brasileira de Computação, Comissão Especial de Bancos de Dados ; organização: UEL, UEM, UTFPR – Campus Cornélio Procópio ; comitê direutivo: Carmem Hara...[et al.]. – Londrina : UEL, 2016.
1 Livro digital.

Tema central: Data Science.

Inclui bibliografia.

Disponível em: <http://cross.dc.uel.br/erbd2016/>

ISSN 2177-4226

1. Banco de dados – Congressos. I. SBC. Comissão Especial de Banco de Dados. II. Universidade Estadual de Londrina. III. Universidade Estadual de Maringá. IV. Universidade Tecnológica Federal do Paraná (Campus de Cornélio Procópio). V. Hara, Carmem Satie. VI. Título. VII. Título: Data science.

CDU 519.68.023

Editorial

É com grande satisfação que apresentamos os artigos aceitos para a décima segunda edição da Escola Regional de Banco de Dados (ERBD) e que compõem os anais do evento. Em 2016, a ERBD foi realizada de 13 a 15 de abril, na cidade de Londrina-PR, envolvendo três instituições na organização: Universidade Estadual de Londrina (UEL), Universidade Tecnológica Federal do Paraná – Campus Cornélio Procópio (UTFPR-CP) e Universidade Estadual de Maringá (UEM). A ERBD é um evento anual promovido pela Sociedade Brasileira de Computação, que tem como objetivo a integração dos participantes, dando oportunidade para a divulgação e discussão de trabalhos em um fórum regional do sul do país sobre bancos de dados e áreas afins. Além das sessões técnicas, a programação do evento inclui oficinas, minicursos e palestras proferidas por pesquisadores de renome da comunidade brasileira.

Mantendo a tradição das edições anteriores da ERBD, foram aceitas submissões de artigos em duas categorias: Pesquisa e Aplicações/Experiências. Todos os artigos foram avaliados por pelo menos 3 membros do Comitê de Programa. A categoria de Pesquisa recebeu 20 submissões, das quais 13 foram aceitas, o que representa 65% de taxa de aceitação. Cada artigo aceito nesta categoria foi apresentado em 20 minutos nas sessões técnicas. A categoria de Aplicações/Experiências recebeu 15 submissões, das quais 6 foram aceitas, o que representa 40% de taxa de aceitação. Artigos desta categoria foram apresentados em 10 minutos nas sessões técnicas, bem como na forma de pôster.

Os *Anais da XII ERBD* representam o resultado do esforço coletivo de um grande número de pessoas. Agradecemos ao Comitê de Organização Local da ERBD, coordenado pelo Prof. Daniel Kaster, que trabalhou arduamente para garantir o bom andamento do evento. Gostaríamos de agradecer também aos membros do Comitê de Programa que fizeram revisões de excelente qualidade. Finalmente, agradecemos aos autores que submeteram seus trabalhos para a ERBD.

Carmem Satie Hara, UFPR
Coordenadora do Comitê de Programa da Categoria Pesquisa

Rebeca Schroeder Freitas, UDESC
Coordenadora do Comitê de Programa da Categoria Aplicações/Experiências

XII Escola Regional de Banco de Dados

13 a 15 de Abril de 2016
Londrina - PR - Brasil

Promoção

Sociedade Brasileira de Computação - SBC

Organização

Universidade Estadual de Londrina - UEL

Universidade Tecnológica Federal do Paraná – Cornélio Procópio - UTFPR-CP

Universidade Estadual de Maringá - UEM

Comitê Diretivo da ERBD

Carmem Satie Hara – UFPR (Presidente)

Fernando José Braz – IFC

Daniel Luis Notari – UCS

Coordenações

Comitê de Programa: Carmem Satie Hara (UFPR)

Palestras: Cristiano R. Cervi (UPF) e Priscila T. Maeda Saito (UTFPR-CP)

Minicursos: Renato Fileto (UFSC) e Raqueline R. de Moura Penteado (UEM)

Oficinas: Nádia Kozievitch (UTFPR-Curitiba), Jacques Duílio Brancher (UEL)

e Helen Cristina de Mattos Senefonte (UEL)

Demos (Aplicações/Experiências): Rebeca Schroeder Freitas (UDESC) e Adilson Luiz Bonifácio (UEL)

Comitê Organizador Local

Daniel dos Santos Kaster – UEL (Coordenador geral)

Adilson Luiz Bonifácio – UEL

Jacques Duílio Brancher – UEL

Helen Cristina de Mattos Senefonte – UEL

Jandira Guenka Palma – UEL (Patrocínios)

Rosana Teixeira Pinto Reis – UEL (Secretaria)

Valdete Vieira Silva Matos – UEL (Secretaria)

Pedro Henrique Bugatti – UTFPR-CP

Alexandre Rossi Paschoal – UTFPR-CP

Priscila Tiemi Maeda Saito – UTFPR-CP

Edson Alves de Oliveira Jr – UEM

Raqueline Ritter de Moura Penteado – UEM

Heloise Manica Paris Teixeira – UEM

Comitê de Programa

Alcides Calsavara - PUCPR
André Luis Schwerz - UTFPR-Campo Mourão
Angelo Fozza - IFC
Carina F. Dorneles -UFSC
Carmem S. Hara - UFPR
Cristiano Cervi - UPF
Daniel Kaster - UEL
Daniel Notari - UCS
Deborah Carvalho - PUCPR
Deise Saccol - UFSM
Denio Duarte - UFFS
Eder Pazinatto - UPF
Edimar Manica - IFF
Edson Ramiro Lucas Filho - UFPR
Eduardo Borges - FURG
Eduardo Cunha de Almeida - UFPR
Fernando José Braz - IFC
Flávio Uber - UEM - UFPR
Guilherme Dal Bianco - UFS
Guillermo Hess - FEEVALE
Gustavo Kantorski - UFSM
Helena Ribeiro - UCS
Jacques Duílio Brancher - UEL
João Marynowski - PUCPR
José Maurício Carré Maciel - UPF
Josiane Michalak Hauagge Dall Agnol - UNICENTRO
Karin Becker - UFRGS
Luiz Celso Gomes Jr - UTFPR
Marcos Aurelio Carrero - UFPR
Marta Breunig Loose - UFSM
Nádia Kozievitch - UTFPR
Priscila Tiemi Maeda Saito - UTFPR
Raquel Stasiu - PUCPR
Raqueline Penteado - UEM - UFPR
Rebeca Schroeder Freitas - UDESC
Regis Schuch - UFSM
Renata Galante - UFRGS
Renato Fileto - UFSC
Ronaldo Mello - UFSC
Sandro Camargo - UNIPAMPA
Scheila de Avila e Silva - UCS
Sergio Mergen - UFSM

Sumário

| | |
|--|-----|
| Artigos Completos de Pesquisa | 8 |
| Artigos Completos de Aplicações/Experiências | 138 |
| Palestras convidadas | 163 |
| Minicursos | 169 |
| Oficinas | 174 |

Artigos Completos de Pesquisa

gos Completos

| | |
|---|----|
| A utilização do Método Cross-Industry Standard Process for Data Mining no Processo de Mineração de Textos: Extração de Termos para Criação de uma Tecnologia Assistiva para o Auxílio à Alunos com Deficiência Motora | 10 |
| <i>Kaio Alexandre da Silva (Universidade de Brasília), Marcos Roberto Pimenta dos Santos, Michel da Silva (Instituto Federal de Educação, Ciência e Tecnologia de Rondônia), Jones Fernando Giaccon (Instituto Federal de Educação, Ciência e Tecnologia de Rondônia), Thyago Borges (Centro Universitário Luterano de Ji-Paraná)</i> | |
| | |
| Uma Avaliação de Algoritmos para Mineração de Dados Disponíveis na WEB .. | 20 |
| <i>Ronaldo Canofre M. dos Santos (Universidade Federal do Rio Grande), Eduardo N. Borges (Universidade Federal do Rio Grande), Karina dos Santos Machado (Universidade Federal do Rio Grande)</i> | |
| | |
| Avaliação de Desempenho de Sistemas Relacionais para Armazenamento de Dados RDF | 30 |
| <i>William Pereira (Universidade do Estado de Santa Catarina), Tiago Heinrich (Universidade do Estado de Santa Catarina), Rebeca Schroeder (Universidade do Estado de Santa Catarina)</i> | |
| | |
| Compressão de Arquivos Orientados a Colunas com PPM | 40 |
| <i>Vinicius F. Garcia (Universidade Federal de Santa Maria), Sergio L. S. Mergen (Universidade Federal de Santa Maria)</i> | |
| | |
| Estratégias para Importação de Grandes Volumes de Dados para um Servidor PostgreSQL | 50 |
| <i>Vanessa Barbosa Rolim (Instituto Federal Catarinense), Marilia Ribeiro da Silva (Instituto Federal Catarinense), Vilmar Schmelzer (Instituto Federal Catarinense), Fernando José Braz (Instituto Federal Catarinense), Eduardo da Silva (Instituto Federal Catarinense)</i> | |
| | |
| Identificação de Contatos Duplicados em Dispositivos Móveis Utilizando Similaridade Textual | 58 |
| <i>Rafael F. Machado (Universidade Federal do Rio Grande), Rafael F. Pinheiro (Universidade Federal do Rio Grande), Eliza A. Nunes (Universidade Federal do Rio Grande), Eduardo N. Borges (Universidade Federal do Rio Grande)</i> | |
| | |
| Implementação de Operadores OLAP Utilizando o Modelo de Programação Map Reduce no MongoDB | 68 |
| <i>Roberto Walter (Universidade Federal da Fronteira Sul), Denio Duarte (Universidade Federal da Fronteira Sul)</i> | |

| | |
|---|-----|
| Mineração de Dados para Modelos NoSQL: um Survey | 78 |
| <i>Fhabiana Thieli dos Santos Machado (Universidade Federal de Santa Maria), Deise de Brum Saccòl (Universidade Federal de Santa Maria)</i> | |
| Mineração de Opiniões em Microblogs com Abordagem CESA | 88 |
| <i>Alex M. G. de Almeida (Universidade Estadual de Londrina), Sylvio Barbon Jr. (Universidade Estadual de Londrina), Rodrigo A. Igawa (Universidade Estadual de Londrina), Stella Naomi Moriguchi (Universidade Federal de Uberlândia)</i> | |
| Um Processo de Avaliação de Dados em um Data Warehouse | 98 |
| <i>Tania M. Cernach (Instituto de Pesquisas Tecnológicas do Estado de São Paulo), Edit Grassiani (Instituto de Pesquisas Tecnológicas do Estado de São Paulo), Renata M. de Oliveira (Centro Paula Souza), Carlos H. Arima (Centro Paula Souza)</i> | |
| Utilizando Técnicas de Data Science para Definir o Perfil do Pesquisador Brasileiro da Área de Ciência da Computação | 108 |
| <i>Gláucio R. Vivian (Universidade de Passo Fundo), Cristiano R. Cervi (Universidade de Passo Fundo)</i> | |
| Workflows para a Experimentação em Análise de Similaridade de Imagens Médicas em um Ambiente Distribuído | 118 |
| <i>Luis Fernando Milano-Oliveira (Universidade Estadual de Londrina), Matheus Peivani Vellone (Universidade Estadual de Londrina), Daniel S. Kaster (Universidade Estadual de Londrina)</i> | |
| XplNet – Análise Exploratória Aplicada a Redes Complexas | 128 |
| <i>Luiz Celso Gomes Jr (Universidade Tecnológica Federal do Paraná), Nádia Kozi evitch (Universidade Tecnológica Federal do Paraná), André Santanchè (Universidade Estadual de Campinas)</i> | |

aper:152887_1

A utilização do método Cross-Industry Standard Process for Data Mining no processo de mineração de textos: extração de termos para criação de uma tecnologia assistiva para o auxílio à alunos com deficiência motora

Kaio Alexandre da Silva¹, Marcos Roberto Pimenta dos Santos, Michel da Silva², Jones Fernando Giacon², Thyago Borges³

¹Departamento de Ciência da Computação – Universidade de Brasília (UnB)
Brasília – DF – Brasil.

²Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO)
Ji-Paraná – RO – Brasil.

³Centro Universitário Luterano de Ji-Paraná (CEULJI/ULBRA)
Ji-Paraná – RO – Brasil.

K4iodm@gmail.com, marcos7947@gmail.com, axel.2k@gmail.com,
jfgiacon@gmail.com, thyago.borges@gmail.com;

Abstract. *The text combines knowledge discovery and extraction techniques of information retrieval, natural language processing and summarization of documents with data mining methods. For dealing with unstructured data, text knowledge discovery is considered more complex than the knowledge discovery in databases. Through articles related to the field of biology, text mining techniques will be applied, combined with the Cross-Industry Standard Process methodology for Data Mining, in order to find the specific words that field of knowledge, thus facilitating decision-making of the words eventually chosen by students with physical disabilities during the writing process.*

Resumo. *A descoberta de conhecimento em texto combina técnicas de extração e de recuperação da informação, processamento da linguagem natural e sumarização de documentos com métodos de Data Mining. Por lidar com dados não-estruturados, a descoberta de conhecimento em texto é considerada mais complexa que a descoberta de conhecimento em base de dados. Através de artigos relacionados ao campo da biologia, serão aplicadas as técnicas de mineração de texto, combinadas com a metodologia Cross-Industry Standard Process for Data Mining, com o objetivo de encontrar as palavras específicas desse campo de conhecimento, facilitando assim a tomada de decisão das palavras a serem escolhidas por alunos com deficiência motora durante o processo de escrita.*

1. Introdução

A preocupação com a Acessibilidade é cada vez maior nas instituições públicas e

privadas. Acessibilidade é um processo de transformação do ambiente e de mudança da organização das atividades humanas, que diminui o efeito de uma deficiência. Para esta transformação do ambiente, para ajudar estas pessoas, a tecnologia é uma aliada poderosa neste processo. As tecnologias que ajudam pessoas com deficiência, são denominada Tecnologias Assistivas (TA).

Dentre as TA's, o uso de computadores para amenizar as dificuldades das pessoas com deficiência é uma das classificações existentes segundo as leis que compõem a ADA (American with Disabilities Act) ADA (1994). O uso de técnicas de Aprendizado Supervisionado e Mineração de Textos (Text Mining) e, documentos textuais são artifícios para a montagem de dicionários controlados em domínios fechados.

Assim este trabalho promoveu o desenvolvimento de um dicionário controlado na área da Biologia utilizando Aprendizado Supervisionado em Documentos Textuais através de Técnicas de Mineração de Textos para serem utilizados no sistema operacional Android, visando auxiliar alunos com deficiência na escrita a melhorar o seu desempenho acadêmico.

A estrutura do artigo foi dividida pela apresentação da problemática, solução proposta, a metodologia, a aplicação da metodologia, as ferramentas utilizadas, teste e avaliação do aplicativo, considerações finais e trabalhos futuros e referências.

1.1. Problemática

Tendo o curso de biologia do Centro Universitário Luterano de Ji-Paraná - CEULJI/ULBRA como objeto de pesquisa, viu-se a dificuldade de escrita de alguns alunos que possuem necessidades especiais, pois várias palavras são de língua estrangeira e os corretores ortográficos não possuem recursos para notificar o aluno da ortográfica exata dessas palavras.

1.2. Solução Proposta

Uma forma de auxiliar os alunos a resolver este problema, principalmente durante as aulas e nos momentos de estudo é disponibilizar um aplicativo que terá um dicionário com as palavras específicas desse conhecimento, possibilitando assim que os alunos consigam anotar com mais facilidade as palavras durante a aula ou durante o seu estudo.

Para se criar esse dicionário, deve-se notar a necessidade da mineração de textos. Sendo que a mineração de texto visa ajudar no processo da extração de conhecimento através de informações semi-estruturadas e não-estruturadas, através de textos, e-mail, artigos, documentos (atas, memorandos, ofícios), dentre outros. A busca de padrões e conhecimento nestes documentos é muito comum. Porém, na maioria das vezes, o resultado obtido é falho: documentos não relacionados, volume muito alto de informações dispensáveis, entre outros.

Através de artigos relacionados ao campo da biologia, serão aplicadas as

técnicas de mineração de texto, com o objetivo de encontrar as palavras específicas desse campo de conhecimento, facilitando assim a tomada de decisão das palavras a serem escolhidas. Considerando que a tomada de decisão é um processo de investigação, de reflexão e de análise, onde tem-se a necessidade de informações qualitativas que contenham alto valor agregado.

2. Metodologia

O processo capaz de gerar conhecimento a partir de dados estruturados nomeia-se de Knowledge Discovery in Database (KDD) ou Descoberta de Conhecimento em Bases de Dados (DCBD). Esse processo combina diversas áreas da descoberta do conhecimento, tais como Aprendizagem de Máquina, Reconhecimento de Padrões, Estatística e Inteligência Artificial, com o objetivo de extraír, de forma automática, informação útil em bases de dados e o Knowledge Discovery in Text (KDT) ou Descoberta de Conhecimento em Texto (DCT) lida com dados não-estruturados. Muitas pesquisas têm sido direcionadas a DCT, por trabalhar com textos, considerada a forma mais natural de armazenamento de informação (Tan, 1999). Para o desenvolvimento do trabalho foi utilizado a técnica de Descoberta de Conhecimento em Texto.

A descoberta de conhecimento em texto combina técnicas de extração e de recuperação da informação, processamento da linguagem natural e summarização de documentos com métodos de Data Mining (DM) Dixon (1997). Por lidar com dados não-estruturados, a descoberta de conhecimento em texto é considerada mais complexa que a descoberta de conhecimento em base de dados. Wives (2000), explica que não se encontram, todavia, metodologias que definam um plano de uso dessas técnicas, completando Loh (2000), relata a lacuna sobre como uma coleção textual deve ser investigada de forma automática ou semi-automática, a fim de que hipóteses sejam validadas.

Magalhães (2002), explica que a Descoberta de Conhecimento em Texto – DCT (Knowledge Discovery in Text – KDT), ao contrário da Descoberta de Conhecimento em Base de Dados, lida com dados não-estruturados. Sendo seu objetivo é extraír conhecimento de bases em que as ferramentas usuais não são capazes de agirem, por não estarem equipadas, ou terem sido desenvolvidas para soluções em dados estruturados.

A metodologia aplicada foi a *Cross-Industry Standard Process for Data Mining* (CRISP-DM), concebida originalmente para mineração de dados. Para a CRISP-DM, o ciclo de vida do processo de DCBD segue uma sequência de etapas CHAPMAN *et al* (2000). Essas etapas são executadas de forma interativa, sendo elas dispostas de acordo com a figura 1.

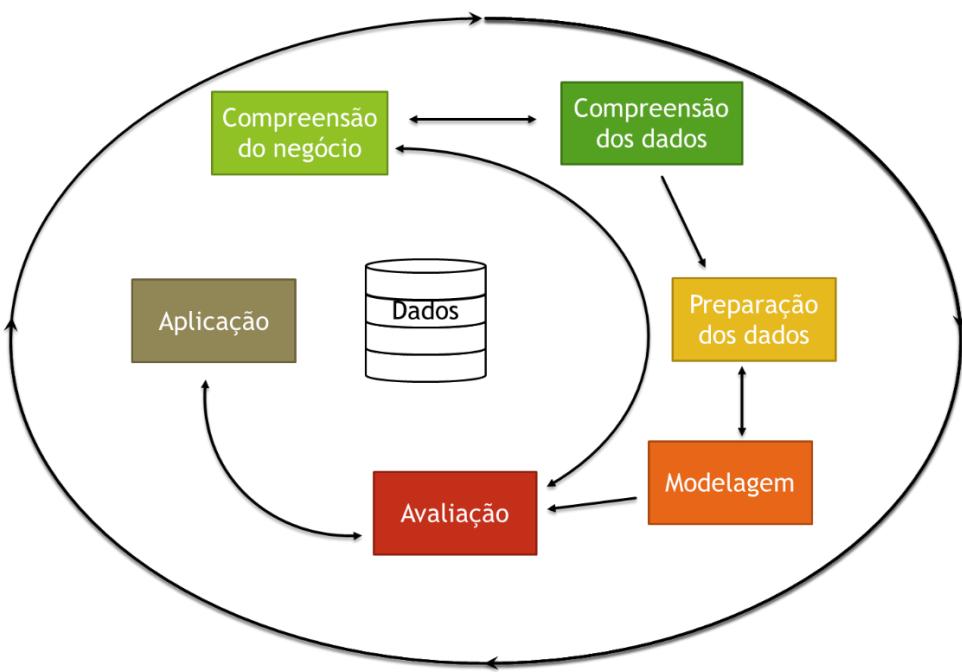


Figura 1 - CHAPMAN, P.; et al. The CRISP-DM Process Model. 2000.

Assim, pelas entradas e respostas providas pelo usuário, a sequência da execução pode ser alterada. O encadeamento das ações, dependendo do objetivo e de como as informações se encontram, permite retorno a passos já realizados.

A Compreensão do Negócio procura identificar as necessidades e os objetivos do negócio do cliente, convertendo esse conhecimento numa tarefa de mineração de dados. Busca detectar eventuais problemas e/ou restrições que, se desconsideradas, poderão implicar perda de tempo e esforço em obter respostas corretas para questões erradas. Essa tarefa compreende ainda descrição do cliente, seus objetivos e descrição dos critérios utilizados para determinar o sucesso do seu negócio.

A Compreensão dos Dados visa a identificar informações que possam ser relevantes para o estudo e uma primeira familiarização com seu conteúdo, descrição, qualidade e utilidade. A coleção inicial dos dados procura adquirir a informação com a qual se irá trabalhar, relacionando suas fontes, o procedimento de leitura e os problemas detectados. Nessa tarefa, descreve-se ainda a forma como os dados foram adquiridos, listando seu formato, volume, significado e toda informação relevante. Durante essa etapa, são realizadas as primeiras descobertas.

A Preparação dos Dados consiste numa série de atividades destinadas a obter o conjunto final de dados, a partir do qual será criado e validado o modelo. Nessa fase, são utilizados programas de extração, limpeza e transformação dos dados. Compreende a junção de tabelas e a agregação de valores, modificando seu formato, sem mudar seu significado a fim de que reflitam as necessidades dos algoritmos de aprendizagem.

Na Modelagem, são selecionadas e aplicadas as técnicas de mineração de dados mais apropriadas, dependendo dos objetivos pretendidos. A criação de um conjunto de

dados para teste permite construir um mecanismo para comprovar a qualidade e validar os modelos que serão obtidos. A modelagem representa a fase central da mineração, incluindo escolha, parametrização e execução de técnica(s) sobre o conjunto de dados visando à criação de um ou vários modelos.

A Avaliação do Modelo consiste na revisão dos passos seguidos, verificando se os resultados obtidos vão ao encontro dos objetivos, previamente, determinados na Compreensão do Negócio, como também as próximas tarefas a serem executadas. De acordo com os resultados alcançados, na revisão do processo, decide-se pela sua continuidade ou se deverão ser efetuadas correções, voltando às fases anteriores ou ainda, iniciando novo processo.

A Distribuição (Aplicação) é o conjunto de ações que conduzem à organização do conhecimento obtido e à sua disponibilização de forma que possa ser utilizado eficientemente pelo cliente. Nessa fase, gera-se um relatório final para explicar os resultados e as experiências, procurando utilizá-los no negócio.

2.1. Aplicação da Metodologia

Sendo a compreensão do negócio o desenvolvimento de um dicionário controlado na área da Biologia utilizando Aprendizado Supervisionado em Documentos Textuais através de Técnicas de Mineração de Textos para serem utilizados no sistema operacional Android, visando auxiliar alunos com deficiência na escrita a melhorar o seu desempenho acadêmico.

Na etapa de compreensão dos dados, artigos foram coletados através da plataforma SciELO, referentes ao tema “Botânica de Fanerograma” e “Botânica de Criptogamas”, que foram escolhidas a partir da matriz curricular do curso de Ciências Biológicas do Centro Universitário Luterano de Ji-Paraná, os temas abordados por essas disciplinas tratam das estruturas de reprodução não se apresentam visíveis (briófitas e pteridófitas) e estruturas de reprodução se apresentam visíveis (gimnospermas e angiospermas). Inicialmente foram coletados cento e cinquenta e sete artigos, porém na fase de preparação de dados alguns artigos só permitiram a visualização do artigo no formato PDF, evitando assim a coleta de dados, todos os artigos tinham título, autor (es), resumo, abstract, desenvolvimento, considerações finais ou conclusão e referências.

Na etapa de preparação dos dados, todos artigos pesquisados foram copiados para a extensão .txt e nomeados em forma ordinal crescente a partir do número “01.txt” até o número “127.txt”. Como parte do processo de preparação dos dados, foi excluído no momento da coleta as partes não relevantes para a pesquisa. Para que não fossem listados foram retiradas as partes que continham autor (es), abstract e referências. Deixando assim apenas o título do artigo, o resumo, o desenvolvimento e a consideração final.

Na etapa da modelagem os artigos foram submetidos a ferramenta de preparação de texto no Text Mining Suite, posteriormente sendo criada uma lista de texto e aplicada a técnica de descoberta por Lista de Conceitos-Chaves, técnica que tem como base a

geração de uma lista com os principais conceitos, com base na frequência existente no texto. Utilizando a função de comparação de texto foi gerada uma lista com doze mil palavras, que foram adotadas como o conjunto de dados que estão em avaliação.

Para a avaliação das palavras foi montado um sistema de avaliação de palavras, foi construído um banco de dados, para armazenar todas as doze mil palavras que foram obtidas na modelagem. Para isso foi criada uma tabela com o nome de “Conjunto”, sua função é apenas guardar as palavras que vem da modelagem. Foi criada mais duas tabelas uma nomeada de “Tipo”, que tem apenas dois registros, que guardam os valores: “1 para Comum” e “2 para Específico”. E a última tabela criada foi nomeada de “Palavras” guarda as palavras e relaciona com o tipo de categoria que o especialista irá determinar. Para essa fase foi estipulado a avaliação das palavras pelos especialistas na área da biologia, que neste trabalho estão sendo representados pelo corpo de professores do curso de Ciências Biológicas do Centro Universitário Luterano de Ji-Paraná.

Como meio de acessar essas palavras no banco de dados foi criado um sistema web utilizando as tecnologias HTML, CSS e PHP. Sistema através do qual o especialista tem acesso a palavra e conta com três opções de ação: Específico, Comum e Deletar. Essas ações foram determinadas no diagrama de classe.

Como resultado da etapa da aplicação, foi construído um aplicativo para a plataforma Android, nomeado de “Palavras da Botânica”, que tem por objetivo adicionar as palavras específicas da botânica ao dicionário do sistema operacional. Possibilitando aos usuários a autocorreção e o auto complemento da palavra no momento da digitação.

A interface do aplicativo é apresentada na figura 2. Nela encontram-se dois botões, o primeiro, “Adicionar Palavras”, adiciona as palavras disponibilizadas pelo aplicativo ao dicionário, atualmente o aplicativo tem duzentas palavras específicas. O segundo botão, “Mostrar Palavras”, abre a lista de palavras que o aplicativo contém, possibilitando ao usuário identificar uma palavra antes de adicionar as mesmas para o dicionário do sistema operacional, ou visualizar quais foram as palavras adicionadas.

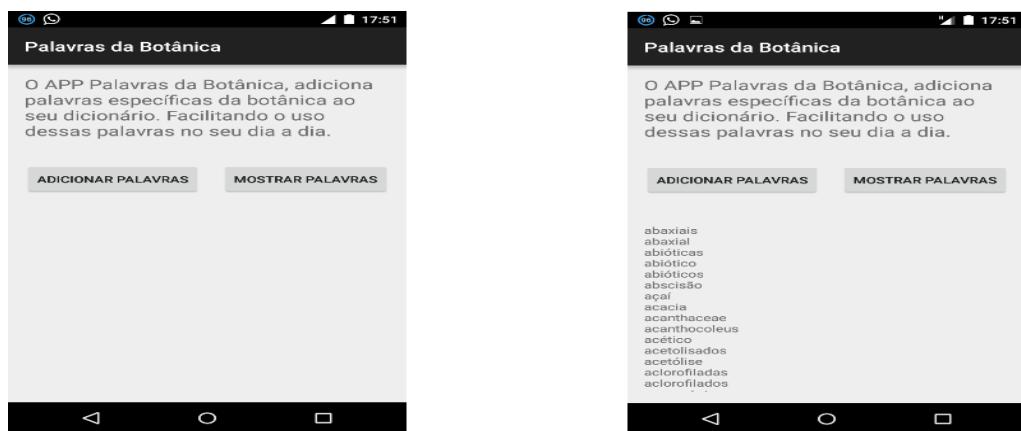


Figura 2 – Interfaces do Aplicativo “Palavras da Botânica”

3. Ferramentas Utilizadas

Para o desenvolvimento do trabalho, foram utilizadas diversas ferramentas, para a coleta dos artigos foi utilizado a plataforma Scientific Electronic Library Online (SciELO), que é uma biblioteca eletrônica que abrange uma coleção selecionada de periódicos científicos brasileiros SCIELO (2002).

Para o processamento dos dados, foi utilizado a ferramenta Text Mining Suite, desenvolvido pela empresa InText Mining, onde a principal técnica do software é a análise de conceitos presentes nos textos INTEXT (2005).

Para a modelagem do diagrama de classe, foi utilizado a ferramenta RAD Studio XE6 e para a modelagem do diagrama entidade-relacionamento, foi utilizado a ferramenta MySQL Workbench 6.2 CE.

Para o desenvolvimento do Sistema de Avaliação de Palavras, foi utilizada a Linguagem de Marcação de Hipertexto (HTML), a Folha de Estilo em Cascata (CSS), a linguagem de programação Hypertext Preprocessor (PHP) e o banco de dados MySQL. Para o desenvolvimento do aplicativo, foi utilizado o ambiente de desenvolvimento do Android Studio.

4. Teste e Avaliação do Aplicativo

O aplicativo foi testado em pesquisa qualitativa, em um ambiente controlado, com dois alunos do curso de Ciências Biológicas, sendo um aluno com deficiência motora e outro sem deficiência. O teste consistiu em digitar duas vezes, uma sequência de cinco palavras específicas da biologia, na primeira vez os alunos utilizaram o celular pessoal, sem o uso do aplicativo, enquanto que na segunda vez os alunos utilizaram as palavras já inseridas no dicionário através do aplicativo.

O aluno com deficiência motora, na primeira etapa levou dois minutos para digitar as cinco palavras, enquanto que na segunda etapa, com o uso do aplicativo, ele levou quarenta e cinco segundos, reduzindo seu tempo em 62,5%.

O aluno sem deficiência, na primeira etapa levou um minuto e vinte e cinco segundos para digitar as cinco palavras, enquanto que na segunda etapa, com o uso do aplicativo, ele levou trinta segundos, reduzindo seu tempo em 64,7%.

Ambos responderam que utilizariam o aplicativo no dia-a-dia e na pergunta sobre a opinião sobre o aplicativo o aluno portador da deficiência complementou que com o uso do aplicativo, aumentaria a eficiência do aprendizado no dia-a-dia.

5. Considerações Finais e Trabalhos Futuros

O presente trabalho visou estabelecer um meio para ajudar as pessoas com deficiência motora a ter um melhor desenvolvimento pedagógico nas disciplinas de biologia, além de focar no estudo da descoberta de conhecimento em texto.

Com os resultados deste estudo pode-se criar dicionários para outras áreas de domínio, ajudando não apenas alunos com deficiência motora, mas sim todos os alunos que estejam estudando aquela área de domínio.

Como foi demonstrado na avaliação do aplicativo, nota-se que o aplicativo não ajuda apenas os alunos com deficiência, mas também os alunos que não possuem nenhum tipo de deficiência e em ambos os casos se notou uma melhora de mais de 60% em relação ao tempo de digitação, o que facilita para o aluno acompanhar o conteúdo que está sendo ministrado e confiança de estar digitando corretamente.

No que diz respeito da aplicação metodológica, obteve total êxito, visto que o projeto de mineração texto foi conduzido pela metodologia CRISP-DM, criada para projetos de mineração de dados. Na prática, verificou-se que não há restrição metodológica para a condução de projetos dessa natureza.

Como trabalhos futuros, pretende-se a finalização da avaliação das palavras, realizar o acompanhamento por um tempo maior dos alunos que estão utilizando o aplicativo a fim de descobrir o impacto no aprendizado, o desenvolvimento de um teclado próprio, para que possa integrar o dicionário ao teclado e ao sugerir as palavras, poder destacar com cores diferentes as palavras específicas das palavras comuns.

Referências

- CAMILO, C. O.; Silva, João Carlos. Mineração de Dados: Conceitos, Tarefas, Métodos e Ferramentas. 2009.
- CAMILO, C. O.; Silva, João Carlos. Um estudo sobre a interação entre Mineração de Dados e Ontologias. 2009.
- CAPUANO, E. A. Mineração e Modelagem de Conceitos como Praxis de Gestão do Conhecimento para Inteligência Competitiva. 2013.
- CAT, 2007. Ata da Reunião VII, de dezembro de 2007, Comitê de Ajudas Técnicas, Secretaria Especial dos Direitos Humanos da Presidência da República (CORDE/SEDH/PR). Disponível em:
<http://www.mj.gov.br/sedh/ct/corde/dpdh/corde/Comite%20de%20Ajudas%20Tecnica s/Ata_VII_Reuniao_do_Comite_de_Ajudas_Tecnicas.doc> Acesso em: 05 maio de 2015.
- CÔRTES, S.C.; LIFSCHITZ, S. Mineração de dados - funcionalidades, técnicas e abordagens. 2002.
- DIXON, Mark. Na Overview of Document Mining Technology. [S.l.: s.n]. 1997.
- EBECKEN, N. F. F.; LOPES, M. C. S.; COSTA, M. C. A. “Mineração de textos”. In: Sistemas inteligentes: fundamentos e aplicação. Barueri Manole. cap. 13, p. 337-370, 2003.
- ELMASRI, Ramez; NAVATHE, Shamkant B. SISTEMA DE BANCO DE DADOS. 6^a EDIÇÃO. Pearson. 2011.

- FERNEDA, Edilson ; PRADO, Hercules A ; SILVA, Edilberto Magalhães . Text Mining for Organizational Intelligence: A Case Study On A Public News Agency. Proceedings 5th International Conference On Enterprise Information Systems Iceis 2003, Angers, França, 2003.
- INTEXT. Manual do Software: Text Mining Suite v.2.4.7. InText Mining Ltda. Versão 10. 2005.
- FIGUEIREDO, C. M. S.; NAKAMURA, E. F. Computação Móvel: Oportunidades e Desafios. T&C Amazônia, v. 1, p. 16-28, 2003.
- GOUVEIA, R. M. M.; GOMES, H. P. ; SOARES, V. G. ; SALVINO, M. M. . Detecção de Perdas Aparentes em Sistemas de Distribuição de Água através de Técnicas de Mineração de Dados.
- HALLIMAN, C. Business intelligence using smart techniques: environmental scanning using text mining and competitor analysis using scenarios and manual simulation. Houston: Information Uncover, 2001.
- KUKULSKA-HULME, A.; TRAXLER, J. Mobile Learning: A handbook for educators and trainers. Routledge, 2005.
- LOH, Stanley ; WIVES, Leandro Krug ; PALAZZO M. de Oliveira, José . Concept-based knowledge discovery in texts extracted from the WEB. SIGKDD Explorations, v. 2, n. 1, p. 29-39, 2000.
- LOH, Stanley; WIVES, Leandro Krug; PALAZZO M. de Oliveira, José. Descoberta proativa de conhecimento em coleções textuais: iniciando sem hipóteses. In: IV Oficina de Inteligência Artificial, 2000, Pelotas. IV OFICINA DE INTELIGÊNCIA ARTIFICIAL (OIA). Pelotas: EDUCAT, 2000. v. 1.
- LOH, Stanley; WIVES, Leandro Krug; PALAZZO M. de Oliveira, José. Descoberta Proativa de Conhecimento em Textos: Aplicações em Inteligência Competitiva. In: III International Symposium on Knowledge Management/Document Management, 2000, Curitiba/PR. ISKM/DM 2000. Curitiba : PUC-PR, 2000. v. 1. p. 125-147.
- MANZINI, E. J. Tecnologia assistiva para educação: recursos pedagógicos adaptados. In: Ensaios pedagógicos: Construindo escolas inclusivas. Brasília: SEESP/MEC, p. 82-86, 2005.
- MATEUS, G. R.; LOUREIRO, A. A. F. Introdução à Computação Móvel. Rio de Janeiro: XI Escola de Computação, 1998. v. 1. p. 189.
- MEIER, Reto. Professional Android Application Development. Indianapolis: Wiley Publishing, 2009.
- MORAIS, E.; AMBRÓSIO, A. P. L. Mineração de Textos. 2007.
- MOULIN, B.; ROUSSEAU, D. Automated knowledge acquisition from regulatory texts. IEEE Expert, Los Alamitos, V.7, n.5, p 27-35, 1992.
- NYIRI, K. Towards a philosophy of m-Learning. In: IEEE INTERNATIONAL WORKSHOP ON WIRELESS AND MOBILE TECHNOLOGIES IN EDUCATION - WMTE, 2002.
- PILTCHER, Gustavo; BORGES, Thyago; LOH, S. ; LITCHNOW, Daniel ; SIMÕES,

- Gabriel. Correção de Palavras em Chats: Avaliação de Bases para Dicionários de Referência. In: Workshop de Tecnologia da Informação e Linguagem, 2005, São Leopoldo. Anais Congresso SBC 2005, 2005. p. 2228-2237.
- PRADO, Hércules A; PALAZZO, J.M.O; FERNEDA, Edilson; WIVES, Leandro K.; SILVA, Edilberto M.S.; LOH, Stanley. Transforming Textual Patterns into Knowledge In RAISINGHANI, Mahesh S. "Business Intelligence in the Digital Economy: Opportunities, Limitations and Risks", CECC, Editora Idea Group, Hershey, PA – EUA, 2003.
- RIBEIRO JR, Luiz Carlos; BORGES, Thyago ; LITCHNOW, Daniel ; LOH, S. ; GARIN, Ramiro Saldaña. Identificação de áreas de interesse a partir da extração de informações de currículos lattes/xml. In: I Escola Regional de Banco de Dados, 2005, Porto Alegre. Anais da I Escola Regional de Banco de Dados. Porto Alegre: UFRGS, 2005. p. 67-72.
- SCIELO. SciELO – Scientific Electronic Library Online. Disponível em: <http://www.scielo.br/scielo.php?script=sci_home&lng=pt&nrm=iso#about>. Acessado em 30 de maio de 2015.
- SILVA, Edilberto M. Descoberta de Conhecimento com o uso de Text Mining: Cruzando o Abismo de Moore. Dissertação de Mestrado em Gestão do Conhecimento e Tecnologia da Informação, UCB, Brasília (DF), dezembro 2002.
- SILVA, Edilberto M.; PRADO, Hercules A; FERNEDA, Edilson. Descoberta de conhecimento com o uso de text mining: técnicas para prover inteligência organizacional. In: VI Jornada de Produção Científica das Universidades Católicas do Centro-Oeste (2002), Goiânia, Set. 2002.
- SILVA, Edilberto M.; PRADO, Hercules A; FERNEDA, Edilson. Suporte à Criação de Inteligência Organizacional em uma Empresa Pública de Jornalismo com o uso de Mineração de Textos. Anais do 3º Workshop Brasileiro de Inteligência Competitiva e Gestão do Conhecimento Congresso Anual da Sociedade Brasileira de Gestão do Conhecimento – KM Brasil 2002, São Paulo-SP, Set. 2002.
- SILVA, Edilberto Magalhães ; PRADO, Hercules Antonio ; OLIVEIRA, Jose Palazzo Moreira de ; FERNEDA, Edilson ; WIVES, Leandro Krug ; LOH, Stanley . Text Mining in the Context of Business Intelligence. In: Mehdi Khosrow-Pour, D.B.A., Information Resources Management. (Org.). Encyclopedia of Information Science and Technology. Hershey, PA: Idea Group, Inc. [ISBN 1-591-40553-X], 2005.
- TAN, A.-H. Text mining: The state of the art and the challenges. Kent Ridge Digital Labs, 1999.
- WIVES, Leandro Krug; PALAZZO, M. de Oliveira, José. Um estudo sobre Agrupamento de Documentos Textuais em Processamento de Informações não Estruturadas Usando técnicas de Clustering. Porto Alegre: CPGCC, 1999.CHAPMAN, P.; CLINTON, J.; KERBER, R.; KHABAZA, T.; REINARTZ, T.; SHEARER, C. & WIRTH, R. CRISPDM 1.0 step-by-step data mining guide. Technical report, CRISP-DM.

aper:152848_1

Uma Avaliação de Algoritmos para Mineração de Dados Disponíveis na Web

Ronaldo Canofre M. dos Santos, Eduardo N. Borges, Karina dos Santos Machado

Centro de Ciências Computacionais – Universidade Federal do Rio Grande (FURG)
Caixa Postal 474, 96203-900, Rio Grande – RS

canofre@inf.ufsm.br, eduardoborges@furg.br, karinaecomp@gmail.com

Abstract. *The limited human capacity to analyze and obtain information from large volumes of data in a timely manner requires the use of techniques and tools of knowledge discovery in databases. This paper presents a review and a study in data mining algorithms implemented in PHP to run on the Web, aiming to demonstrate the feasibility of using these tools for data mining task.*

Resumo. *A limitada capacidade humana de analisar e obter informações a partir de grandes volumes de dados em um tempo hábil exige a utilização de técnicas e ferramentas de descoberta de conhecimento em banco de dados. Este trabalho apresenta uma revisão e um estudo sobre algoritmos de mineração de dados implementados em PHP para execução na Web, visando demonstrar a viabilidade da utilização destas ferramentas para tarefa de mineração de dados.*

1. Introdução

O vasto volume de dados gerados até os dias de hoje já ultrapassa a marca de 4,4 ZB (zettabytes) segundo pesquisa da EMC divulgada em 2014 e é acrescido diariamente tanto pela interação homem/máquina como por equipamentos que não necessitam da intervenção humana. Nesta mesma pesquisa é estimado ainda que em 2020 já tenham sido gerados cerca de 44 ZB ou 44 trilhões de gigabytes [EMC Corporation 2014].

Segundo Han et al. (2011), atualmente não existe mais a era da informação, mas sim a era dos dados. Volumes gigantescos são gerados e armazenados diariamente por inúmeras áreas, tais como ciência, medicina, comércio e engenharia. Transações financeiras, vendas online, pesquisas e experimentos realizados, registros médicos e sistemas de monitoramento são alguns exemplos de como eles são gerados.

No entanto, essa grande quantidade de dados passa a ter valor a partir do momento em que se torna possível a extração de conhecimento a partir deles, o que é realizado pelo processo de Descoberta de Conhecimento em Banco de Dados ou KDD (*Knowledge Discovery in Databases*). Neste processo, a Mineração de Dados (MD) é a principal etapa e consiste na combinação de métodos tradicionais de análise de dados com algoritmos sofisticados para processamento de grandes volumes [Tan et al. 2006].

Para realização do processo de KDD, em especial a etapa de mineração de dados, existem inúmeras ferramentas disponíveis, tanto livres como comerciais. Em geral, essas ferramentas necessitam ser instaladas em um equipamento para que

executem localmente, como por exemplo: Weka, R, ODM, MDR, KMINE, Pimiento, dentre outras [Camilo and da Silva 2009].

Este trabalho tem como objetivo realizar uma revisão e um estudo sobre algoritmos de mineração de dados que executem diretamente em um ambiente Web, sem a necessidade de instalação de programas, bibliotecas e pacotes. Mais especificamente, este artigo revisa ferramentas desenvolvidas na linguagem PHP¹.

Dessa forma, a principal contribuição deste trabalho é a busca e avaliação de ferramentas que possam ser utilizadas sem a necessidade de instalação e independentes de Sistema Operacional (SO), facilitando assim a portabilidade e o acesso. Soma-se a estes benefícios a possibilidade de permitir e facilitar o aprendizado prático da Mineração de Dados agilizando, por exemplo, a sua utilização em sala de aula.

O restante deste texto está organizado como segue. A fundamentação teórica sobre KDD e Mineração de Dados está descrita na seção 2. A seção 3 apresenta uma abordagem geral sobre ferramentas de mineração de dados, a metodologia e as bases de dados utilizadas. Na seção 4 são apresentados os resultados, incluindo as avaliações realizadas e, por fim, na seção 5 são apresentadas as conclusões e propostas para trabalhos futuros.

2. Fundamentação Teórica

2.1. *Knowledge Discovery in Databases*

A possibilidade da aplicação de processos de KDD em todos os tipos de dados fortalece a sua utilização para inúmeras aplicações, tais como: gerenciamento de negócios, controle de produção, análise de mercado, pesquisas científicas, dentre outros [Han et al. 2011].

Segundo Fayyad et al. (1996), KDD consiste em um processo não trivial de identificação de novos padrões válidos, potencialmente úteis e compreensíveis, aplicado sobre um conjunto de dados, visando melhorar o entendimento de um problema ou auxiliar na tomada de decisão (Figura 1). Também pode ser classificado também como um processo interativo, iterativo, cognitivo e exploratório, englobando vários passos e tomada de decisões por parte dos analistas envolvidos.

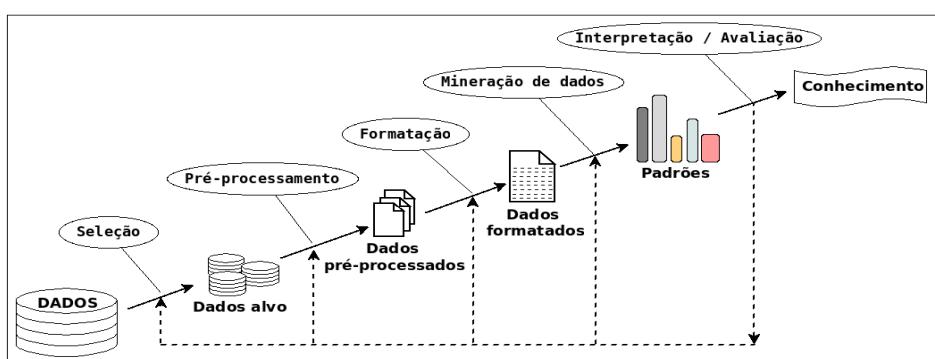


Figura 1. Processo de KDD adaptado de Fayyad et al. (1996).

¹ Linguagem interpretada utilizada para desenvolvimento de páginas Web.

De acordo com Han et al. (2011), esse processo é dividido nas fases de limpeza, integração, seleção, transformação, mineração, avaliação e apresentação do conhecimento, as quais são divididas em etapas de pré e pós-processamento:

- pré-processamento – etapa mais demorada e trabalhosa do processo de KDD e consiste na preparação dos dados disponíveis na sua forma original a fim de se tornarem apropriados para análise. No entanto, mesmo sendo trabalhosa, é uma etapa importante, pois possibilita a obtenção de melhores resultados quanto ao tempo, custo e qualidade [Tan et al. 2006].
- pós-processamento – etapa realizada após a mineração dos dados (abordada na seção 2.2), realizando a interpretação dos padrões minerados e a implantação do conhecimento, assegurando que apenas resultados válidos e úteis sejam incorporados a ferramentas de apoio a decisão.

2.2 Mineração de Dados

Segundo Han et al. (2011), Fayyad et al. (1996) e Tan et al.(2006), a mineração de dados consiste em uma etapa no processo de KDD que busca descobrir ou obter padrões a partir de um grande volume de dados, os quais representem informações úteis. Sua realização consiste na aplicação de algoritmos que se utilizam de técnicas estatísticas e de inteligência artificial, para realizar a classificação de elementos de um conjunto de dados e prever valores de variáveis aleatórias.

É importante ressaltar a diferença entre uma tarefa e uma técnica de mineração. As regularidades ou categorias de padrões que se deseja encontrar com a busca realizada, são especificadas nas tarefas, ao passo que as técnicas de mineração especificam os métodos que possibilitam descobrir os padrões desejados [Bueno and Viana 2012].

As tarefas realizadas pela mineração de dados, podem ser divididas entre preditivas, ou supervisionadas, e descritivas, ou não supervisionadas [Tan et al 2006; Mamon e Rokach 2010]. As tarefas descritivas, como associação, agrupamento e sumarização, realizam a busca de padrões com base na correlação entre os dados. Já as tarefas preditivas, como classificação e regressão, tem o objetivo de prever o valor de um determinado atributo baseado no valor de outros.

Embora essas técnicas ou métodos sejam classificados de acordo com a tarefa realizada, essa separação segue uma linha tênue, visto que alguns métodos preditivos podem realizar tarefas descritivas e vice-versa [Fayyad et al. 1996]:

- Agrupamento ou Clustering – consiste em técnicas de aprendizado não supervisionado que a partir de um conjunto de dados gera subgrupos baseados na semelhança dos objetos [Han et al. 2011];
- Classificação – processo realizado em duas etapas: uma de aprendizado ou treinamento, onde o modelo de classificação é construído a partir de um conjunto de registros com rótulos conhecidos e a etapa de classificação onde o modelo é utilizado para prever os rótulos de determinado conjunto de dados [Han et al. 2011];
- Associação – consiste em identificar relacionamentos interessantes escondidos em grandes volumes de dados, os quais podem ser representados na forma de regras de associação ou conjuntos de itens frequentes, podendo ser

expressas no formato SE condição ENTÃO resultado [Tan et al. 2006];

- Regressão – técnica de modelagem preditiva cuja variável alvo é um valor contínuo², principal característica que a diferencia da técnica de classificação. Pode ser utilizada, por exemplo, para prever o índice da bolsa de valores com base em outros indicadores econômicos ou a idade de um fóssil baseado na quantidade de carbono-14 presente [Tan et al. 2006].

3. Ferramentas e algoritmos para mineração de dados

Algumas aplicações executáveis utilizadas para mineração de dados, como Knime, Orange Canvas, Rapidminer Studio e Weka, implementam mais de um método e/ou algoritmo para realização das tarefas de mineração, apresentando ainda uma interface gráfica para sua utilização [Boscarioli et al. 2014].

Já outras aplicações podem implementar somente um método e/ou algoritmo, o que em geral pode ser observado em implementações para fins específicos tais como comparações de desempenho ou propostas de novos algoritmos. A revisão realizada neste trabalho a cerca de ferramentas online para mineração de dados focou em implementações de ferramentas/algoritmos na linguagem PHP, para uso diretamente no navegador em qualquer sistema operacional, sem necessidade de instalação.

A definição pela linguagem PHP foi baseada em características como configuração do servidor, utilização de complementos, custos de hospedagem e desempenho. A instalação e configuração de servidores web para páginas PHP, ocorre de maneira mais direta e simplificada [Alecrim 2006 and Sverdlov 2012] e requer um menor número de configurações quando comparada a instalação de um servidor para páginas desenvolvidas em Java [Pinto 2015 and Feijo 2015].

Ainda, o custo de hospedagem para servidores com suporte a JSP³, é relativamente superior aos servidores para linguagem PHP, o que pode ser verificado em empresas como KingHost e Locaweb, o que se justifica devido ao maior consumo de memória e processamento utilizado por esta linguagem [CppCMS 2015].

As aplicações encontradas durante a revisão e selecionadas para avaliação concentram-se nas técnicas de agrupamento e classificação, sendo observado com mais ênfase implementações dos algoritmos K-means para técnicas de agrupamento e k-Nearest Neighbor (k-NN), baseado na técnica de classificação. Tais ferramentas foram selecionadas devido à possibilidade de alteração da base de dados a ser analisada e por não apresentarem erros de desenvolvimento durante a execução.

A metodologia a ser empregada na análise das aplicações consiste em obter bases de dados para cada técnica, realizando posteriormente a avaliação de seus resultados. Tais fontes foram obtidas do UC Irvine Machine Learning Repository [Linchman 2013] e são utilizadas como entrada nas respectivas aplicações e na ferramenta Weka e realizando posteriormente a avaliação e comparação dos resultados.

² Valores contínuos se referem a uma infinita possibilidade de representações dentro de um intervalo.

³ JavaServer Pages, linguagem para desenvolvimento de páginas HTML baseada na linguagem Java.

4. Resultados obtidos

Nos algoritmos encontrados durante a revisão bibliográfica realizada, os dados de entrada são informados diretamente no código, através de arquivos CSV⁴ e/ou vetores, ou seja, essas implementações analisadas, não apresentam interface gráfica. As saídas desses algoritmos são apresentadas no navegador com poucas informações e sem tratamento visual, dificultando a análise dos resultados. As configurações dos parâmetros são também realizadas mediante edição de variáveis e métodos das classes, sendo adotadas em sua maioria as licenças GPL, LGPL e MIT.

4.1 Implementações do algoritmo K-means

O algoritmo de agrupamento K-means utiliza um método de particionamento baseado em protótipos que busca encontrar um determinado número de grupos (k), definidos pelo usuário [Tan et al. 2006]. Os grupos são definidos em torno de um centroide, o qual é a média de um grupo de pontos e em geral não consiste em um valor da base de dados.

Após os grupos formados, um novo centroide é definido com base nos dados deste novo grupo e os grupos são redistribuídos. O algoritmo é finalizado quando os pontos permanecerem no mesmo grupo ou os centroides não sofrerem alteração. Foram estudadas três implementações do algoritmo K-means, definidas como Kmeans01 [Delespierre 2014], Kmeans02 [Yokoyama 2011] e Kmeans03 [Roob 2014].

A saída de cada implementação estudada foi modificada de forma a padronizar o resultado em todas as implementações, com a finalidade de facilitar o entendimento e as avaliações, exibindo assim os centroides de cada cluster, identificados pelo seu índice numérico, seguido da quantidade de objetos do grupo no seguinte formato: Cluster Y [x1,x2,x3,...,xn]: N points.

As avaliações foram realizadas considerando o número de grupos, número de instâncias em cada grupo e como os resultados são exibidos. Como os algoritmos avaliados não implementam nenhuma métrica de validação de agrupamento (como por exemplo DBI, Sillhouette, C-index, etc.) [Tomasini et al. 2016], as mesmas não foram utilizadas para avaliar a qualidade dos resultados obtidos com cada algoritmo.

Com relação ao número de grupos, as implementações K-means encontradas permitem a definição de $k > n$, não respeitando a premissa de que a quantidade de clusters definidas deve ser menor ou igual à quantidade de objetos existentes ($k \leq n$), permitindo assim a ocorrência de grupos vazios. A aplicação Kmeans02 trata estes grupos, transformando o elemento mais distante de qualquer um dos centroides em um cluster e, neste caso, a quantidade de clusters gerados é inferior ao definido em k .

A inicialização dos centroides é realizada de forma randômica nas implementações, sendo possibilitado pela aplicação Kmeans01 uma variação denominada K-means++ que permite uma inicialização alternativa do agrupamento [Arthur and Vassilvitskii 2007] e pela kmeans02, limitado entre o valor mínimo e máximo presente na base de dados. A condição de parada adotada consiste na

⁴ CSV - *Comma Separated Values*, formato de arquivo de dados tabelados separados por vírgulas

estabilidade dos centroides, ou seja, quando os mesmos não sofrem mais alteração, sendo tal informação obtida através da exibição dos passos do algoritmo.

Com relação aos atributos, ocorre apenas a restrição ao número de dimensões da base de dados, pela implementação Kmeans03. A implementação Kmeans01 [Delespierre 2014] permite um número ilimitado de atributos, segundo sua documentação, sendo a maior base utilizada, composta por 33 atributos. As alterações no valor de k refletem no total de agrupamentos vazios, na igualdade da distribuição das instâncias.

A implementação Kmeans02 [Yokoyama 2011] necessitou que os limites de tempo e uso de memória do servidor fossem alterados, quando utilizada a base Turkiye Student Evaluation, sendo tais alterações realizadas através das funções set_time_limit e init_set [PHP Documentation Group 2015], as quais respectivamente alteram o tempo limite de execução e o valor para quantidade de memória a ser utilizada pelo servidor.

Por fim, a implementação Kmeans03 [Roob 2014] suporta somente agrupamentos formados por duas dimensões, sendo por este motivo avaliada com a base de dados disponibilizada no próprio algoritmo, o qual é composto por 19 coordenadas de um plano cartesiano, entre as coordenadas (0,0) e (20,20). Dessa forma, considerando a quantidade reduzida de objetos possível de ser analisada, foram também utilizadas poucas variações de valores para k, sendo observado o surgimento mais constante de clusters vazios para os valores mais elevados. Devido a essa restrição, essa implementação não foi executada com as bases de dados do UCI.

4.1.1 Comparação Dos Resultados Das Implementações K-means

Para os resultados das implementações Kmeans01 e Kmeans02, as quais permitiam alterar a base de dados a ser analisada, foram utilizadas além das mesmas bases de dados, um valor único para k. Buscando complementar a avaliação realizada, as bases de dados foram também utilizadas na implementação do K-means na ferramenta Weka.

A Tabela 1 apresenta o resultado o total de instâncias em cada grupo, referente a execução de cada algoritmo e do aplicativo Weka, para cada uma das bases de dados, com k=5, sendo as bases de dados identificadas como segue: 01-seeds, 02-Wholesale customers e 03-Turkiye Student Evaluation.

Tabela 1. Resultados das implementações Kmeans01, Kmeans02 e Weka.

| Base de dados | Instâncias/ Atributos | Instâncias em cada grupo por Implementação | | |
|---------------|--------------------------|--|---------------------------------|--------------------------------|
| | | Kmeans01 | Kmeans02 | Weka |
| 01 | 210/7 | [42,64,16,46,42] | [49, 49, 55, 42, 15] | [14, 46, 50, 48, 52] |
| 02 | 440/8 | [113,63,23,6,235] | [235, 113, 63, 6, 23] | [239, 98, 8, 36, 59] |
| 03 | 5820/33 | [1342, 901, 1140, 1261, 1176] | [1344,1175, 902, 1140, 1259] | [760, 731, 1971, 1622, 736] |

Devido as aplicações analisadas não implementarem métricas de validação, e a inicialização dos centroides ser realizada de forma aleatória, gerando *clusters* distintos a cada execução, os resultados apresentados foram obtidos a partir de uma sequência de 10 execuções, sendo selecionados os mais recorrentes. Dessa forma, é possível avaliar que os resultados das implementações Kmeans01 e Kmeans02 apresentam uma mesma quantidade de instâncias ou quantidades idênticas, tais como os [...] e [...1140,]

para as bases 01 e 03 respectivamente e uma mesma distribuição para base 02.

Quando comparados com os resultados do Weka, também foram encontrados *clusters* com o mesmo número de instâncias, como o valor [..46,] presente nos resultados da base 01 para o Kmeans01 e o Weka, além de valores muito próximos. Cabe ressaltar que as variações de resultados observadas durante as execuções das aplicações Kmeans01 e Kmeans02 não foram percebidas durante a execução do Weka.

4.2 Implementações do algoritmo k-NN

O método de classificação baseado no vizinho mais próximo utiliza a técnica de descoberta baseada em instância, a qual requer uma medida de proximidade para classificar os objetos, sendo geralmente utilizado no reconhecimento de padrões [Lima 2011]. Como os objetos são representados como pontos por um classificador, a forma mais comum de determinar as proximidades é através do cálculo da distância entre os pontos. Neste trabalho foram analisadas três implementações do algoritmo k-NN: Knn01 [Degerli 2012], Knn02 [Mafrur 2012] e Knn03 [Houweling 2012].

Essas implementações apresentam similaridades entre suas características, destacando-se a utilização do método Euclíadiano adotado para obtenção das distâncias e a definição da instância a ser classificada, que não apresenta restrições fixas com relação a novos valores. As demais características individuais são abordadas a seguir.

A implementação denominada como Knn01 [Degerli 2012] é a mais limitada dentre as analisadas, restringindo o número de atributos a 2 e fixando a definição de vizinhos mais próximos ($k=4$) e as classes, diretamente no código fonte. Dessa forma, as avaliações para esta aplicação se baseiam somente na base de dados disponibilizada. As informações de distâncias e de vizinhos mais próximos, somente são visualizadas através da impressão de um vetor pré-formatado, não permitindo a recuperação dos dados para uma exibição mais intuitiva.

A implementação Knn02 [Mafrur 2012], restringe a 4 o número de atributos da base de dados, não incluindo a classe a qual o objeto pertence, permitindo a definição do número de vizinhos (k) a ser utilizada. A classificação das instâncias é realizada através da posição da mesma na base de dados, sendo assim necessário incluir uma nova instância na base para que a mesma possa ser classificada. Esta implementação necessitou ainda que os limites de tempo de execução do servidor fossem alterados de seu valor padrão, quando utilizada a base banknote authentication. Tais alterações foram realizadas através do uso da função `set_time_limit` [PHP Documentation Group 2015], definido um novo tempo limite de execução.

A última implementação do algoritmo k-NN analisada, Knn03 [Houweling 2012], não apresenta restrições quanto a quantidade de atributos, permitindo a definição de um peso para cada atributo, conforme proposto por Paredes e Vital (2006). A classificação realizada por esta aplicação, ocorre de forma distinta com relação as demais implementações k-NN, não definindo o número de vizinhos mais próximo a serem analisado (k), utilizando para tal, a média das distâncias de cada classe para definir a classe de um novo objeto.

4.2.1 Comparação Dos Resultados Das Implementações k-NN

Visando realizar uma comparação entre os resultados das implementações Knn02 e Knn03, os quais possibilitam a execução com uma mesma base de dados, foram realizados testes com um mesmo conjunto de instâncias a serem classificadas, sendo definidos para um valor de $k=5$ e o peso dos atributos de Knn03 foi mantido com o valor 1, com o objetivo de não influenciar nos resultados.

Ainda, buscando complementar a avaliação realizada, foram também analisadas através da aplicação Weka, utilizando as mesmas definições adotadas nas aplicações analisadas. Para realização dos testes foram utilizadas instâncias já classificadas, existentes nas bases de dados, permitindo assim verificar a acurácia das classificações realizadas. Assim sendo, os resultados obtidos demonstraram um comportamento similar entre os resultados das aplicações, onde as mesmas realizaram tanto classificações corretas como incorretas para todas as instâncias analisadas.

A Tabela 2 demonstra a relação de instâncias e atributos existentes em cada base de dados, bem como o conjunto de testes utilizados e os resultados obtidos com cada implementação, através dos quais é possível verificar por exemplo, que a implementação Knn03 apresentou uma média de acerto melhor, sendo para esse pequeno conjunto de teste mais efetivo do que a implementação do Weka.

Tabela 2. Resultados das implementações Knn02 e Knn03 e Weka.

| Base de Dados | Instância/ Atributo | Atributos da instância de teste | Classificação Correta | Knn02 | Knn03 | Weka |
|----------------|---------------------|----------------------------------|-----------------------|-------------------|-------------------|-------------------|
| 01 | 748/4 | (1,24,6000,77) | Não | Não | Não | Sim |
| | | (4,6,1500,22) | Sim | Não | Sim | Sim |
| 02 | 1372/4 | (-0.4928,3.060, -1.8356,-2834) | Verdadeira | Verdadeira | Verdadeira | Verdadeira |
| | | (0.6636,-0.0455, -0.1879,0.2345) | Verdadeira | Verdadeira | Falsa | Verdadeira |
| 03 | 150/4 | (4.9,2,0.4,0.1,7) | Virginica | Versicolor | Virginica | Versicolor |
| | | (5.9,3.2,4.8,1.8) | Versicolor | Virginica | Virginica | Virginica |
| Acurácia média | | | | 50% | 66% | 50% |

5. Conclusão e trabalhos futuros

A utilização de aplicações *online* tem almejado facilitar o cotidiano dos usuários, abstraindo questões presentes em aplicações *desktop*⁵ e também favorecendo a portabilidade entre os sistemas operacionais, dentre outros fatores. Dessa forma, a utilização de ferramentas *Web* para descoberta de conhecimento possibilita a realização destas tarefas, independente de instalação e sistema operacional, favorecendo o uso em ambientes de aprendizagem por exemplo.

A análise inicial das implementações encontradas levou em consideração os dados de entrada, a apresentação da saída, a forma de uso da aplicação e a licença utilizada, sendo selecionadas para uma avaliação mais detalhada, as aplicações que possibilitam a modificação dos dados de entrada e a execução através do navegador.

⁵Ferramentas instaladas no computador.

Devido à inexistência de interface gráfica, as configurações foram realizadas diretamente no código fonte.

A realização de testes com bases de dados obtidas no repositório UCI, seguindo uma configuração padrão para as execuções e instâncias cujas classes eram previamente conhecidos no caso do k-NN. Os resultados gerados foram comparados também com os obtidos pelo *software* Weka, buscando uma maior confiabilidade nos resultados.

Foi possível observar um comportamento comum entre as saídas de todas as aplicações, as quais apresentaram tanto classificações corretas e incorretas no caso do k-NN e para o K-means, resultados muito próximos e em alguns casos iguais, em relação ao número de instâncias em cada agrupamento. No entanto, as avaliações realizadas possibilitaram alcançar os objetivos iniciais propostos por este trabalho, demonstrando a ausência de ferramentas *web* completas desenvolvidas em PHP que possibilitem uma utilização similar as ferramentas *desktop*, permitindo também enumerar inúmeras aplicações únicas de algoritmos, desenvolvidas em PHP, as quais possibilitam a utilização por meio de navegadores.

Os testes realizados também contribuíram com a comprovação da capacidade de utilização de ferramentas de mineração de dados de forma *online* demonstrando um correto funcionamento e sem custos de processamento, possibilitando ainda a utilização multiplataforma.

Assim sendo, a pesquisa realizada demonstra uma vasta área de estudos, proporcionando a realização de trabalhos futuros, tanto na área de desenvolvimento como pesquisa. Algumas implementações possíveis seriam de aplicações individuais, tais como as analisadas, no entanto, de forma mais intuitivas, com interface gráfica e prontas para uso. Também a elaboração de uma ferramenta completa composta por vários algoritmos, tal como as aplicações *desktop* existentes.

Referências

- Alecrim, E. (2006). Conhecendo o Servidor Apache (HTTP Server Project). em: <http://www.infowester.com/servapach.php>. Acesso em: novembro de 2015.
- Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pages 1027–1035. Society for Industrial and Applied Mathematics.
- Boscarioli, C., Viterbo, J. and Teixeira, M. F. (2014). Avaliação de aspectos de usabilidade em ferramentas pra mineração de dados. Anais da I Escola Regional de Sistemas de Informação do Rio de Janeiro, 1(1):107-114.
- Bueno, M. F. and Viana, M. R. (2012). Mineração de dados: aplicações, eficiência e usabilidade em ferramentas para mineração de dados. Anais do congresso de Iniciação Científica do INATEL, 1(1):86–95.
- Camilo, C. O. and da Silva, J. C. (2009). Mineração de Dados: Conceitos, tarefas métodos e ferramentas. Relatório técnico, Instituto de Informática. Universidade Federal de Goiás, Goiânia.
- CppCMS. (2015). CppCMS: The C++ Web Development Framework. Disponível em: http://cppcms.com/wikipp/en/page/benchmarks_all. Acesso em: novembro de 2015.

- Degerli, O. (2012). Data Mining Algorithms' Application with PHP. Disponível em: <https://github.com/onurdegerli/data-mining>. Acesso em: setembro de 2015.
- Delespierre, B. (2014). PHP K-Means. Disponível em: <https://github.com/bdelespierre/php-kmeans>. Acesso em: setembro de 2015.
- EMC Corporation (2014). The digital universe of opportunities: Rich data and the increasing value of the internet of things. Disponível em: <http://brazil.emc.com/leadership/digital-universe/2014iview/executive-summary.htm>. Acesso em: agosto de 2015.
- Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996). Advances in knowledge discovery and data mining. chapter From Data Mining to Knowledge Discovery: An Overview, pages 1–34. American Association for Artificial Intelligence, Menlo Park, CA, USA.
- Feijó, D. (2015). Instalando e configurando o Apache TomCat no seu servidor Linux. Disponível em: <http://daniellfeijo.com/2015/08/17/instalando-e-configurando-o-apache-tomcat-no-seu-servidor-linux/>. Acesso em: novembro de 2015.
- Han, J., Kamber, M. and Pei, J. (2011). Data Mining Concepts and Techniques. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 3^a edition. São Francisco. USA.
- Houweling, F. (2012). Knn prototype php. Disponível em: <https://github.com/FrankHouweling/KnnPrototypePhp>. Acesso em: setembro de 2015.
- Lima, G. F. (2011). Classificação Automática de Batidas de Eletrocardiogramas. Trabalho de graduação, Curso de Ciência da Computação, Instituto de Informática. Universidade Federal do Rio Grande do Sul/RS.
- Mafrur, R. (2012). Knn php. Disponível em: https://github.com/rischanlab/knn_php. Acesso em: setembro de 2015.
- Paredes, R. and Vidal, E. (2006). Learning Weighted Metrics to Minimize Nearest-Neighbor Classification Error, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(7):1100-1110.
- PHP Documentation Group (2015). Manual do PHP. Disponível em: http://php.net/manual/pt_BR/. Acesso em: outubro de 2015.
- Roob, S. (2014). Php k-means. Disponível em: <https://github.com/simonrobb/php-kmeans>. Acesso em: setembro de 2015.
- Tan, P., Steinbach, M., and Kumar, V. (2006). Introduction to Data Mining. Pearson international Edition. Pearson Addison Wesley.
- Tomasini, C., Emmendorfer, L., Borges, E. and Machado, K. A methodology for selecting the most suitable cluster. In: ACM/SIGAPP Symposium on Applied Computing, 2016 (to appear).
- Yokoyama, S. (2011). K-means php. Disponível em: <https://github.com/abarth500/K-Means-PHP>. Acesso em: setembro de 2015.

aper:152921_1

Avaliação de Desempenho de Sistemas Relacionais para Armazenamento de dados RDF

William Pereira¹, Tiago Heinrich¹, Rebeca Schroeder¹

¹Departamento de Ciências da Computação – Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas – 89.219-710 – Joinville – SC – Brasil

{willpereirabr,tiagoheinrich1995}@gmail.com, rebeca.schroeder@udesc.br

Abstract. Nowadays, an increasing amount of data are becoming available in RDF format. In order to provide suitable database systems to manage RDF data, there are approaches adapting relational database systems to process RDF, or using NoSQL systems to deal with the large volume of some RDF datasets. This paper presents an experimental study which compares two models of relational systems in this context. The first model, named triple-store, is a simple solution that converts an RDF dataset to a relation. The second model is based on the RDF structure to provide a more appropriate relation schema. These models are represented in the experiments by the systems Jena-TDB Fuseki and ntSQL, respectively. The results reported that the relational schema applied by ntSQL provides better response time in SPARQL queries than compared to the schema of a triple-store.

Resumo. Atualmente, observa-se um volume crescente de dados sendo publicado no formato RDF. Para prover sistemas de bancos de dados adequados para gerenciar este tipo de dados, as soluções partem de adaptações dos SGBDs relacionais para suportar RDF, assim como sistemas NoSQL para suprir a demanda do volume de algumas bases de dados deste tipo. Este artigo apresenta um estudo experimental que compara dois modelos de sistemas relacionais neste contexto. O primeiro destes modelos, conhecido como triple-store, é uma solução simples que transforma uma fonte RDF em uma tabela. O segundo modelo utiliza noções da estrutura RDF para propor um esquema relacional mais robusto. Estes modelos são representados nos experimentos pelos sistemas Jena-TDB Fuseki e o ntSQL, respectivamente. Os resultados obtidos demonstram que o uso de um esquema relacional mais robusto, como o obtido pelo ntSQL, confere um melhor desempenho em consultas SPARQL submetidas a estes repositórios.

1. Introdução

RDF (*Resource Description Framework*) é hoje o modelo padrão para representação de dados na Web (Apache Jena 2016a). A partir da estrutura de identificadores da Web, o RDF permite definir semanticamente os relacionamentos entre dados através do uso de URIs (*Universal Resource Identifier*). Dados RDF são definidos através de triplas, compostas de um sujeito, um objeto e relacionados por um predicado. Com a padronização, diversas fontes passaram a produzir dados em RDF continuamente. Como resposta a esta realidade, bases de dados de diferentes tamanhos e características estão disponíveis neste

formato, em especial, na Web. Algumas destas fontes de dados estão disponíveis no site *Large Triple Stores*¹.

O crescente volume de dados no formato RDF criou a necessidade por sistemas de bancos de dados capazes de gerenciar dados neste modelo. Em geral, sistemas NoSQL ou repositórios de grande escala têm sido adotados para o armazenamento de fontes RDF em virtude do elevado volume de dados que algumas fontes apresentam (Zeng et al. 2013). Entretanto, o uso de sistemas deste tipo acrescentam uma maior complexidade ao desenvolvimento de aplicações pois, em geral, estes sistemas não apresentam algumas das características de um Sistema Gerenciador de Banco de Dados (SGBD) relacional (Arnaut et al. 2011). Desta forma, para repositórios com volumes de dados de dimensões convencionais o uso de SGBDs relacionais tem sido preferidos por alguns trabalhos.

No contexto de SGBDs Relacionais que suportam RDF, existem dois tipos de modelo aplicados. O primeiro, bastante simples, é conhecido como *triple-store*. Um *triple-store* define um banco RDF através de uma única relação composta pelos campos sujeito, predicado e objeto. Nesta relação, as tuplas correspondem a triplas RDF. Exemplos de sistemas que empregam este modelo são o Jena TDB (Apache Jena 2016a) e RDF-3X (Neumann and Weikum 2010). O segundo modelo corresponde à utilização de conhecimentos sobre a estrutura RDF para definição de um esquema relacional baseada em tipos. Neste caso, cada tipo representa uma relação da base de dados. O sistema ntSQL (Bayer et al. 2014) é um dos sistemas que emprega este modelo.

Existem diversos trabalhos, como J. Huang, D. Abadi 2011, Ravindra et al. 2011 e Papailiou et al. 2014, que provam a ineficiência de *triple-store*, especialmente no desempenho de consultas RDF mais complexas. Esta ineficiência é devida ao tamanho atingido pela relação do *triple-store*, e do custo das auto-junções necessárias para o desempenho de consultas RDF complexas. Segundo Bayer et al. 2014, a ausência de conhecimento sobre a estrutura dos dados RDF faz com que diversas soluções utilizem SGBDs relacionais em sua forma mais simples através de um *triple-store*. Entretanto, a ausência de um esquema acaba por sub-utilizar o modelo relacional ao criar relações baseadas apenas nas composições de triplas. Conforme apontado por Pham 2013, apesar de RDF constituir um modelo livre de esquema, é possível a extração de estruturas de dados a partir de diversas fontes RDF. Esta possibilidade viabiliza uma representação RDF mais adequada em SGBDs relacionais, bem como um melhor desempenho em consultas. Neste contexto, este trabalho visa investigar a diferença no desempenho em consultas de um *triple-store* com um banco equivalente que aplique o segundo modelo através do ntSQL.

Este artigo tem por objetivo apresentar um estudo experimental que compara o desempenho em consultas utilizando os dois tipos de modelos aplicados por SGBDs Relacionais para RDF. Este estudo compara o *triple-store* Jena TDB Fuseki (Apache Jena 2016a) como representante do primeiro modelo, e o ntSQL (Bayer et al. 2014) como representante do segundo modelo. Os experimentos foram baseados no *Belin SPARQL Benchmark* (Bizer and Schultz 2009) através de consultas SPARQL e seu gerador de bases de dados. Os resultados obtidos apontam um ganho significativo no desempenho de consultas RDF utilizando o modelo empregado pelo ntSQL, comparado ao *triple-store*.

¹<https://www.w3.org/wiki/LargeTripleStores>

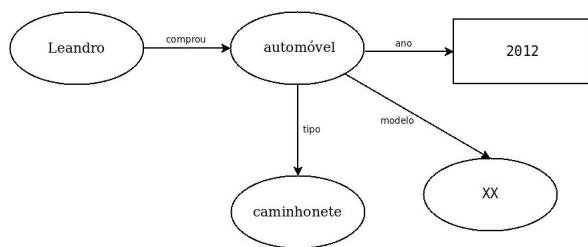


Figura 1. Representação gráfica de dados RDF.

Este trabalho está organizado em mais 5 seções. A Seção 2 apresenta o modelo RDF e sua linguagem de consulta denominada SPARQL. A seção seguinte introduz os modelos de sistemas relacionais para armazenamento de dados RDF. Adicionalmente, é apresentado o *triple-store* Jena-TDB, bem como o sistema ntSQL que é o representante dos sistemas que utilizam noções da estrutura de dados RDF. A Seção 4 apresenta a avaliação experimental realizada por este trabalho, que por sua vez compara o *triple-store* Jena TDB-Fuseki com o ntSQL. Os trabalhos relacionados a este trabalho, no que se refere a outras avaliações similares, são apresentados pela Seção 5. A Seção 6 apresenta as conclusões deste trabalho, bem como as perspectivas de trabalhos futuros.

2. Conceitos Iniciais

Esta seção introduz conhecimentos necessários para a compreensão do estudo experimental a ser apresentado por este artigo. Para tanto, as seções a seguir apresentam o modelo RDF e sua linguagem de consulta SPARQL.

2.1. RDF

RDF é a sigla para *Resource Description Framework* e que, segundo a *World Wide Web Consortium*, é um *framework* para representação de informações na Web. Uma característica básica do modelo RDF é que os metadados utilizados para descrever características de um site, por exemplo, precisam seguir uma estrutura básica de organização. Esta estrutura é reconhecida como tripla, composta por *sujeito-predicado-objeto*.

Para demonstrar este aspecto, considere como exemplo a seguinte afirmação: “Leandro comprou um automóvel.” Neste caso, *Leandro* é o sujeito, *comprou* é o predicado e *automóvel* é o objeto. Esta frase, ou tripla, pode ser representada com sua estrutura sujeito-predicado-objeto através de um grafo. A Figura 1 mostra um grafo composto por esta e outras triplas. Embora o exemplo fornecido omita este detalhe, observa-se que por ser um modelo de dados voltado à Web, as informações que denotam sujeitos e objetos são especificadas por identificadores de recursos na Web dados por URIs (*Uniform Resource Identifier*). Ou seja, um sujeito ou objeto poderia ser, ao invés de um nome, uma URI para um site que tenha informações sobre o dado. Na representação gráfica, as elipses representam sujeitos e objetos especificados por URIs, as setas representam predicados e os retângulos representam objetos que são do tipo literal.

Com base no exemplo da Figura 1 é possível extrair as seguintes triplas do grafo direcionado: “Leandro comprou automóvel”, “automóvel tipo caminhonete”, “automóvel ano 2012”, “automóvel modelo XX”.

2.2. SPARQL

SPARQL (**S**PARQL **P**rotocol **a**nd **R**D**F** **Q**uery **L**anguage) é uma linguagem de consulta sobre dados RDF. Ela define consultas através de padrões de triplas RDF na forma de *sujeito-predicado-objeto*. Com base na Figura 1, um exemplo de utilização da linguagem SPARQL é verificar o modelo e ano do automóvel comprado por Leandro através da seguinte consulta:

```
1      SELECT DISTINCT ?qualmodelo, ?qualano
2      WHERE {
3          Leandro comprou automovel
4          automovel modelo ?qualmodelo
5          automovel ano ?qualano
6      }
```

Observe que a estrutura de formação da consulta se dá por padrões de triplas. Os elementos das triplas podem ser especificados conforme um grafo RDF, ou serem definidos como variáveis utilizando o ? como prefixo. No exemplo os objetos que referem-se ao modelo e cor do automóvel são tratados como variáveis cujos valores serão obtidos e retornados pela consulta.

SPARQL suporta uma variedade de consultas. Entretanto, a linguagem possui limitações como, por exemplo, sub-expressões não são suportadas. A abrangência do SPARQL aumenta conforme a utilidade do RDF também aumenta, e com isso, a necessidade de *benchmarks* para testar as capacidades da linguagem. Um *benchmark* para este cenário é o *Berlin SPARQL Benchmark* ou BSBM. O objetivo do BSBM, segundo Bizer and Schultz 2009, é ajudar os desenvolvedores a encontrar a melhor arquitetura e o melhor sistema de banco de dados para suas necessidades.

O BSBM é baseado em um caso de uso de um sistema de *e-commerce*, onde uma lista de produtos é oferecida por vendedores e posteriormente avaliada por clientes através de revisões. Um exemplo de consulta SPARQL do BSBM é dada a seguir:

```
1
2      PREFIX rdf: <http://www.w3.org/.../22-rdf-syntax-ns#>
3      PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4      PREFIX bsbm:<http://www4.wiwiss.fu-berlin.de/.../>
5      SELECT ?product ?label
6      WHERE {
7          ?product rdfs:label ?label .
8          ?product rdf:type bsbm:Product .
9          FILTER regex(?label, "%word1%")
10     }
```

Esta consulta é uma das mais simples encontradas no BSBM, e visa obter produtos que tem como rótulo uma *string* específica. Processos de *benchmarking* utilizando o BSBM foram executados sobre diversos sistemas de armazenamento que suportam RDF. Como mencionado anteriormente, o foco do presente artigo está sobre sistemas relacionais para armazenamento RDF. Para tanto, a seção a seguir apresenta alguns modelos empregados por estes sistemas e suas principais características.

3. Sistemas Relacionais para Armazenamento RDF

Nesta seção são apresentados dois modelos de armazenamento de dados RDF em sistemas relacionais, bem como alguns dos sistemas que os implementam. Os sistemas apresentados são comparados na avaliação a ser apresentada pela Seção 4.

3.1. Triple-Stores

Um dos formatos para armazenamento RDF, o *Triple-Store* utiliza uma mistura do formato de um modelo de sistema gerenciador de banco de dados relacional com a facilidade de inferência de dados do modelo RDF. Em um *triple store*, dados RDF são armazenados como um conjunto de triplas em uma tabela Abadi et al. 2009. O diferencial deste formato é a facilidade de criação da base de dados, para tal só é necessário criar uma única tabela que irá conter três campos, esses campos serão respectivamente o campo do sujeito, do predicado e do objeto. Deste modo, as tuplas desta tabela correspondem às triplas de um grafo RDF.

Existem alguns exemplos de sistemas que utilizam este formato de armazenamento. Em geral, estes sistemas suportam outros modelos de armazenamento fornecendo portabilidade quanto a diferentes tipos de dados. Alguns exemplos de *triple stores* são o Virtuoso (W3C 2016), Jena TDB (Apache Jena 2016b) e o RDF3-X (Neumann and Weikum 2010). Dentre estes, apenas o RDF3-X (Neumann and Weikum 2010) suporta exclusivamente o modelo RDF. O Jena TDB Fuseki (Apache Jena 2016a) é uma extensão do Jena TDB com suporte exclusivo a consultas RDF.

O Jena TDB é um componente do projeto Jena que fornece armazenamento e consultas para modelos utilizados na Web Semântica, como OWL, RDF e XML. Foi desenvolvido para atuar como um repositório de dados para a Web Semântica de alto desempenho, mesmo em uma única máquina. Seu sistema de armazenamento utiliza o modelo de *triple-store* apresentado anteriormente. O processador de consultas SPARQL é servido pelo componente Jena Fuseki Apache Jena 2016a, que por sua vez é considerado um servidor SPARQL. Em conjunto com o TDB, o Fuseki provê um sistema de armazenamento persistente e transacional para RDF. O Jena Fuseki é um *framework* Java de código aberto.

3.2. ntSQL

O ntSQL é uma ferramenta para a conversão de bases de dados RDF para bases de dados relacionais. A ferramenta é composta de dois módulos. O primeiro módulo compreende um conversor de dados RDF em formato NT para *scripts* de criação de esquemas relacionais, bem como instruções para a inserção de dados no formato SQL Bayer et al. 2014. O segundo módulo da ferramenta corresponde ao mapeamento de consultas SPARQL para consultas SQL sobre o esquema relacional produzido pelo primeiro módulo. Embora em operação, este segundo módulo não se encontra disponível para publicação.

O mapeamento de dados RDF para o modelo relacional é baseado na extração da estrutura RDF a partir de suas fontes de dados. Em resumo, assume-se que sujeitos e objetos não-literais estão relacionados a seus respectivos tipos. No mapeamento, após identificados os tipos, relações para cada tipo são criadas tendo como tuplas os dados relacionados aos respectivos tipos no grafo RDF. Relacionamentos entre os tipos são também identificados para a devida criação de chaves estrangeiras entre as relações criadas.

```
< Usuario1 > < type > < Pessoa >.  
< Usuario1 > < nome > < Pedro >.  
< Usuario2 > < type > < Pessoa >.  
< Usuario2 > < nome > < Joao >.  
< Usuario3 > < type > < Pessoa >.  
< Usuario3 > < nome > < Maria >.  
< Usuario1 > < responsavelPor > < Usuario2 >.  
< Usuario1 > < responsavelPor > < Usuario3 >.
```

Tabela 1. Triplas RDF

Como exemplo de mapeamento RDF-Relacional, considere o seguinte conjunto de triplas RDF dadas no formato NT pela Tabela 1. No formato NT cada tripla é representada por uma linha, sendo que sujeito, predicado e objeto são colocados entre <>. O mapeamento obtido pelo ntSQL para este conjunto de triplas pode ser representada pela relação da Tabela 2. Como pode ser observado nas triplas apresentadas, os usuários são todos do tipo *Pessoa*, e podem estar relacionados entre si através do predicado *responsavelPor*. Neste caso, o mapeamento para relacional é dado pela criação de uma relação para este tipo *Pessoa*, sendo que o campo *id* pode ser definido como a chave primária da relação, e o campo *responsavel* como uma chave estrangeira representando o relacionamento entre os usuários.

Embora o exemplo apresentado seja simples, é possível perceber que a extração de estruturas de dados a partir de tipos RDF viabiliza uma representação RDF mais adequada em SGBDs relacionais, se comparada ao modelo *triple-store*. A comparação entre estes dois modelos é estabelecida pelos experimentos da próxima seção.

4. Avaliação Experimental

Nesta seção é apresentada uma avaliação experimental que compara o desempenho de um sistema do tipo *triple-store* com o ntSQL, isto é, dois sistemas relacionais destinados ao armazenamento de dados RDF. Os sistemas comparados correspondem aos sistemas apresentados pelas Seções 3.1 e 3.2 deste artigo, isto é, TDB (Fuseki) e o ntSQL respectivamente. A métrica utilizada por esta avaliação refere-se ao desempenho destes sistemas dado pelo tempo de resposta em consultas SPARQL. As bases utilizadas por este experimento, bem como as consultas, foram extraídas do *benchmark* Berlin SPARQL Benchmark (BSBM) introduzido pela Seção 2.2.

Para a escolha dos sistemas de armazenamento comparados, foi escolhido o sistema TDB (Fuseki) por ser o triple-store do projeto Jena, que por sua vez aparece no topo do *ranking* do site DB-Engines (DB-Engines 2016) como sistema de armazenamento mais popular e destinado ao modelo RDF. Em princípio o sistema RDF-3X também havia sido escolhido devido a sua evidência como sistema de referência em diversos artigos. No entanto, observou-se que o sistema foi descontinuado e a versão mais recente disponível apresentava alguns *bugs*, bem como alguns resultados inconsistentes. Quanto ao ntSQL, sua escolha se deu pelo caráter inovador de seu modelo de armazenamento, se comparado aos *triple-stores*. O sistema de gerenciamento de banco de dados utilizado pelo ntSQL foi o MySQL. Recomenda-se a leitura da Seção 3.2, para uma melhor compreensão do modelo aplicado pelo ntSQL.

As configurações aplicadas no experimento, bem como os resultados obtidos, são apresentados pelas seções a seguir.

Tabela 2. Relação Pessoas

| id | nome | responsavel |
|-----------|-------------|--------------------|
| Usuario1 | Pedro | |
| Usuario2 | João | Usuario1 |
| Usuario3 | Maria | Usuario1 |

Tabela 3. Número de triplas RDF por quantidade de produtos

| Triplas RDF | Produtos |
|-------------|----------|
| 100 | 40382 |
| 200 | 75555 |
| 400 | 156054 |
| 800 | 297721 |
| 1600 | 585208 |
| 2000 | 725830 |

4.1. Configurações do Experimento

A máquina utilizada para os testes possui 4G de memória RAM, um processador AMD phenom II x4 e o sistema operacional linux-ubuntu 14.04. Para a realização do experimento, foram utilizadas as consultas 1, 6 e 9 do BSBM. Entre as 11 consultas SPARQL disponibilizadas pelo BSBM escolheu-se estas 3 pois representam tamanhos diferentes de consultas em termos da quantidade de padrões de triplas apresentadas por cada um, bem como da quantidade de resultados retornados. Desta forma, acredita-se conferir uma abrangência representativa do BSBM na comparação dos sistemas e seus resultados. Para verificar a escalabilidade dos sistemas comparados utilizaram-se bases com tamanhos variados. No BSBM o fator de escala da base corresponde a quantidade de produtos do sistema de *e-commerce*. No caso do experimento foram utilizadas bases de 100, 200, 400, 800, 1600 e 2000 produtos. Uma base com 2000 produtos possui 725830 triplas de RDF, como pode ser observado na Tabela 3. A comparação entre os 3 sistemas escolhidos é apresentada na seção a seguir com base na métrica do tempo de resposta para as consultas do BSBM.

4.2. Resultados obtidos

Como mencionado na seção anterior, as três consultas escolhidas do BSBM foram utilizadas para os testes em seis tamanhos de base diferentes. Cada consulta foi executada dez vezes para cada tamanho de base, dos quais foram retirados para cada base a média e mediana. Os tempos de resposta dos 2 sistemas comparados com relação as consultas 1, 3 e 9 do BSBM são apresentados pelas Figuras 2, 3 e 4, respectivamente.

Observa-se em todas as consultas que o Fuseki apresenta um desempenho muito inferior em comparação com o ntSQL. O ntSQL provou ter um melhor tempo para efetuar as consultas, demonstrando um desempenho com um crescimento quase constante, não possuindo nenhuma variação drástica com o crescimento do tamanho do banco. Em relação ao Fuseki seu crescimento apresenta picos de acordo com o crescimento do tamanho dos bancos, com aumento do tempo de resposta muito superior ao ntSQL. Acredita-se que esta diferença possa ser explicada pelo modelo de armazenamento utilizado por *triple-stores* ao processar consultas com diversos padrões de triplas. Ao colocar todas as triplas em uma mesma tabela, além de gerar grandes arquivos para as relações, surge a necessidade da execução de auto-junções para a recuperação dos diversos padrões de triplas das consultas. Por exemplo, para o ntSQL na Consulta 1 foram necessárias 2 junções, enquanto que no Fuseki ocorreu um total de 4 auto-junções. No caso desta consulta, a diferença nos tempos de resposta dos sistemas pode ser explicada pela quantidade de junções ou auto-junções necessárias. Entretanto, verificou-se que o tamanho da relação do *triple-store* também determina o desempenho nas consultas. Por exemplo, na Consulta 3, não houveram junções para o ntSQL, e apenas 1 auto-junção para o Fuseki. Na Consulta

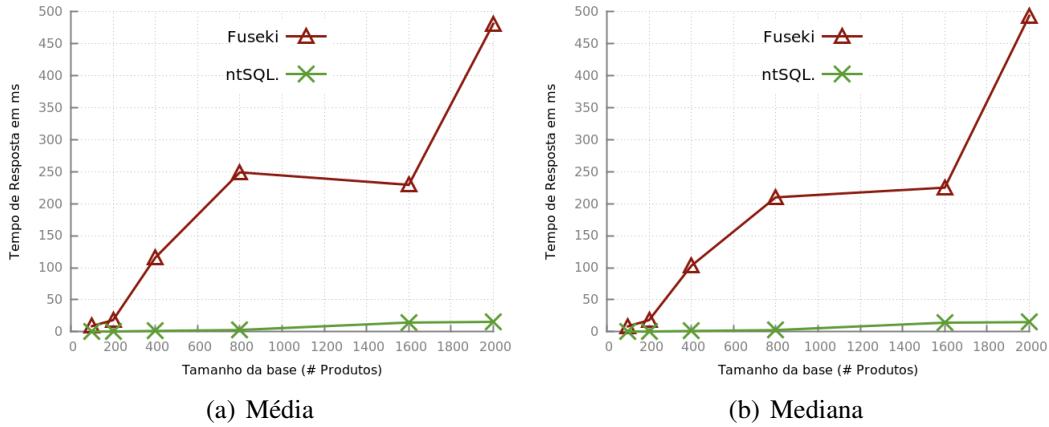


Figura 2. Desempenho dos Sistemas - Consulta 1 do BSBM

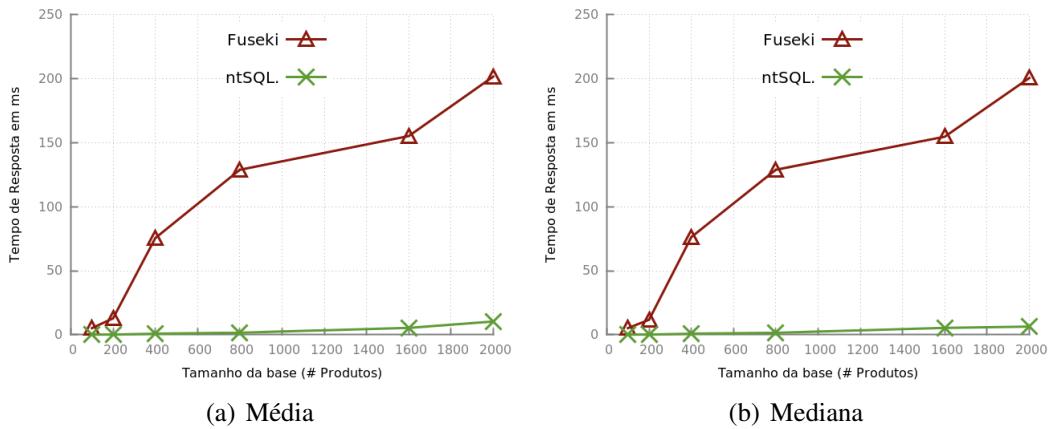


Figura 3. Desempenho dos Sistemas - Consulta 3 do BSBM

9, nenhuma junção ou auto-junção foi necessária para ambos os sistemas, o que demonstra o impacto do tamanho da relação do *triple-store* na recuperação de dados. Assim, verifica-se que o modelo empregado pelo ntSQL mostra-se mais adequado neste sentido por distribuir os dados em diferentes relações, agrupando-os de acordo com seus tipos.

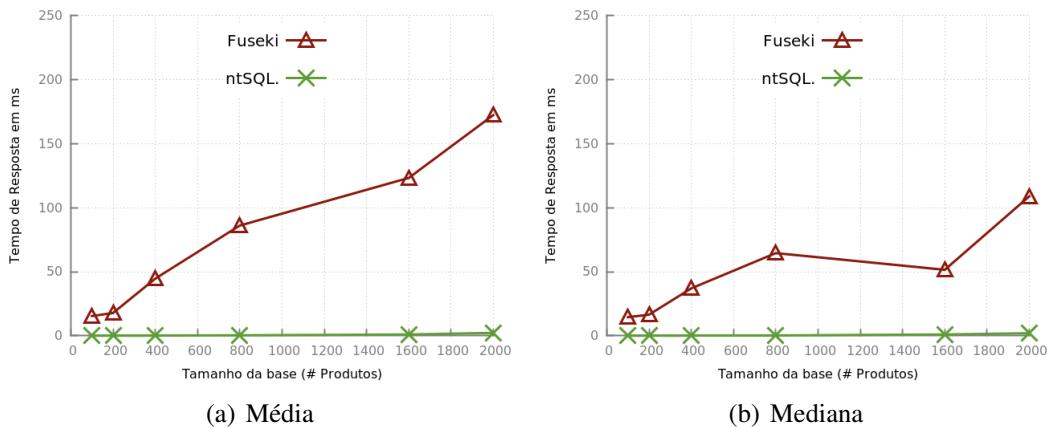


Figura 4. Desempenho dos Sistemas - Consulta 9 do BSBM

5. Trabalhos Relacionados

Nesta seção será discutido os resultados de outros trabalhos que apresentam avaliações referentes a sistemas para RDF, mostrando assim o desempenho médio de bancos triple-store quando comparados entre si e também entre outros modelos de armazenamento. O primeiro trabalho escolhido é o *TDB Results for Berlin SPARQL Benchmark* (Bizer and Schultz 2009), que utiliza o *benchmark* do Berlin para avaliar o Jena-TDB. Foram feitos testes com 50k, 250k, 1M, 5M, 25M e 100M triplas do Berlin SPARQL Benchmark (BSBM). O TDB utilizado por este trabalho difere do que foi utilizado pelo estudo experimental apresentado pela Seção 4 por utilizar um outro processador de consultas SPARQL diferente do Fuseki, além de sua avaliação se tratar de um ambiente distribuído. Apesar das diferenças, observou-se que os tempos apresentados pelo presente trabalho são proporcionais aos obtidos nas respectivas consultas avaliadas por este trabalho.

Além deste, outro trabalho relacionado ainda com o BSBM é o *BSBM with Triples and Mapped Relational Data* (Orri Erling 2016). Este trabalho tem como objetivo demonstrar que um esquema relacional mapeado a partir de um RDF tem um poder de processamento melhor, com respostas mais rápidas para as consultas do que um *triple-store*. O banco utilizado para os testes foi o OpenLink Virtuoso, que é considerado o *triple-store* mais rápido dentre os disponíveis no mercado (Morsey et al. 2011). Para uma base de 100M triplas no sistema, considerando o conjunto de consultas (*query mix*) original do BSBM, o *triple-store* avaliado por este trabalho conseguiu executar 5746 QMpH (*Query Mix Per Hour*), enquanto que para o repositório relacional mapeado do RDF o total foi de 7525 QMpH. Este resultado por si só mostra uma superioridade do repositório relacional mapeado, assim como constatado pelo presente trabalho através do ntSQL.

Outro trabalho que vale destaque é o *DBpedia SPARQL Benchmark – Performance Assessment with Real Queries on Real Data* (Morsey et al. 2011), que tem como objetivo criar e testar um novo *benchmark* com um caso de uso real, dados reais e aplicabilidade já testadas na Web Semântica. Além deste caso real, os testes foram realizados em três sistemas RDF de evidência no cenário atual, ou seja, que estão bem rankeados no DB-Engines 2016. Eles são o Virtuoso, Sesame e BigOWLIM. Os dados carregados no sistema são extraídos da DBpedia, uma grande diferença em relação aos outros *benchmarks* que possuem dados sintéticos. O *dataset* escolhido por Morsey et al. 2011 possui 153.737.776 triplas. Os resultados do trabalho mostram que o Virtuoso tem o melhor desempenho entre os três em mais de 90% dos casos, e o segundo melhor sistema é o BigOWLIM seguido do Sesame. Esses dois últimos tem resultados bem próximos, porém o BigOWLIM tem vantagem nos maiores *datasets*, enquanto o Sesame tem vantagem nos menores.

6. Conclusão

Este artigo apresentou um estudo experimental que compara dois modelos de armazenamento de dados RDF em sistemas relacionais através dos sistemas Jena-TDB Fuseki e ntSQL. Como esperado, o Jena-TDB Fuseki apresentou um desempenho inferior ao ntSQL, atestando as desvantagens do uso de *triple-stores* já apontadas por outros trabalhos. O melhor desempenho do ntSQL pode ser atribuído ao uso de um esquema relacional mais robusto do que os utilizados por *triple-stores*. Como trabalho futuro, pretende-se envolver outros sistemas na avaliação e outros *benchmarks*. Além disto, pretende-se de-

senvolver uma análise mais detalhada que possa justificar o desempenho dos sistemas através de características de consultas e do esquema de banco de dados.

Agradecimentos: Este trabalho foi parcialmente suportado pelos programas de iniciação científica PIC&DTI e PIPES da Universidade do Estado de Santa Catarina.

Referências

- Abadi et al. 2009 Abadi, D. J., Marcus, A., Madden, S. R., and Hollenbach, K. (2009). SW-Store: A Vertically Partitioned DBMS for Semantic Web Data Management. *The VLDB Journal*, 18(2):385–406.
- Apache Jena 2016b Apache Jena (Acesso em Fevereiro de 2016b). Apache Jena TDB. <https://jena.apache.org/documentation/tdb/index.html>.
- Apache Jena 2016a Apache Jena (Acesso em Janeiro de 2016a). Apache Jena Fuseki. <https://jena.apache.org/documentation/fuseki2/>.
- Arnaut et al. 2011 Arnaut, D., Schroeder, R., and Hara, C. (2011). Phoenix: A Relational Storage Component for the Cloud. In *IEEE International Conference on Cloud Computing (CLOUD)*, pages 684–691.
- Bayer et al. 2014 Bayer, F. R., Nesi, L. L., and Schroeder, R. (2014). ntSQL: Um Conversor de Documentos RDF para SQL. In *Anais da Escola Regional de Banco de Dados*. SBC.
- Bizer and Schultz 2009 Bizer, C. and Schultz, A. (2009). The Berlin SPARQL Benchmark. In *International Journal on Semantic Web & Information Systems*.
- DB-Engines 2016 DB-Engines (Acesso em Janeiro de 2016). DB-Engines Ranking of RDF Stores. <http://db-engines.com/en/ranking/rdf+store>.
- J. Huang, D. Abadi 2011 J. Huang, D. Abadi, K. R. (2011). Scalable SPARQL Querying of Large RDF Graphs. *PVLDB*, 4(11):1123–1134.
- Morsey et al. 2011 Morsey, M., Lehmann, J., Auer, S., and Ngomo, A.-C. N. (2011). DBpedia SPARQL Benchmark – Performance Assessment with Real Queries on Real Data. In *International Semantic Web Conference*.
- Neumann and Weikum 2010 Neumann, T. and Weikum, G. (2010). The rdf-3x engine for scalable management of rdf data. *The VLDB Journal*, 19(1):91–113.
- Orri Erling 2016 Orri Erling (Acesso em Janeiro de 2016). BSBM with Triples and Mapped Relational Data. <http://www.openlinksw.com/dataspace/doc/oerling/weblog/Orri>
- Papailiou et al. 2014 Papailiou, N., Tsoumakos, D., Konstantinou, I., Karras, P., and Koziris, N. (2014). H2rdf+: An efficient data management system for big rdf graphs. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD ’14, pages 909–912. ACM.
- Pham 2013 Pham, M. (2013). Self-organizing structured RDF in MonetDB. In *IEEE 29th International Conference on Data Engineering Workshops (ICDEW)*, pages 310–313.
- Ravindra et al. 2011 Ravindra, P., Hong, S., Kim, H., and Anyanwu, K. (2011). Efficient processing of rdf graph pattern matching on mapreduce platforms. In *Proceedings of the Second International Workshop on Data Intensive Computing in the Clouds*, DataCloud-SC ’11, pages 13–20. ACM.
- W3C 2016 W3C (Acesso em Janeiro de 2016). OpenLink Virtuoso. https://www.w3.org/2001/sw/wiki/OpenLink_Virtuoso.
- Zeng et al. 2013 Zeng, K., Yang, J., Wang, H., Shao, B., and Wang, Z. (2013). A Distributed Graph Engine for Web Scale RDF data. *Proceedings of the VLDB Endowment*, 6(4):265–276.

aper:152861_1

Compressão de Arquivos Orientados a Colunas com PPM

Vinicio F. Garcia¹, Sergio L. S. Mergen¹

¹Universidade Federal de Santa Maria
Santa Maria – RS – Brasil

vfulber@inf.ufsm.br, mergen@inf.ufsm.br

Abstract. Column oriented databases belong to a kind of NoSQL database in which the values of the same column are stored contiguously in secondary memory. This physical organization favors compression, mainly because the proximity of data of the same nature decreases the information entropy. With respect to high cardinality columns that store text, several compression methods can be used. One of them, called PPM, is usually good in obtaining high compression rates, but the execution time is poor for conventional files. The purpose of this paper is to analyze whether this compression method is able to explore the nature of column oriented data to obtain more expressive results in comparison with its main competitors.

Resumo. Bancos de dados orientados a colunas pertencem a um tipo de banco NoSQL em que os valores de uma mesma coluna são armazenados contiguamente em memória secundária. Essa organização física favorece a compressão, uma vez que a aproximação dos dados de mesma natureza diminui a entropia da informação. Considerando especificamente colunas de alta cardinalidade que armazenam texto, diversos tipos de compressores podem ser usados. Um deles, chamado PPM, costuma obter boas taxas de compressão, mas possui um tempo de processamento considerado alto para arquivos convencionais. O objetivo desse artigo é verificar se esse método consegue explorar a natureza dos dados orientados a coluna de forma a obter resultados mais expressivos em relação aos seus concorrentes.

1. Introdução

Os bancos de dados NoSQL tem recebido muita atenção recentemente. Ao contrário dos bancos de dados relacionais, essa nova vertente utiliza diferentes formas de organização de arquivos. Utilizando arquiteturas baseadas na computação em nuvem, esse tipo de solução oferece um bom escalonamento para determinados tipos de aplicações que usam padrões de acesso aos dados bem específicos.

Um dos tipos de banco NoSQL que se popularizou é conhecido como orientados a colunas (Han et al., 2011). Diferentemente dos SGBDs convencionais que armazenam registros de tabelas consecutivamente em arquivos, os sistemas orientados a colunas armazenam todos os valores de uma mesma coluna consecutivamente, possivelmente em arquivos separados, conforme ilustrado na Figura 1.

Essa forma de organização é útil em alguns cenários específicos, como por exemplo, para acelerar a execução de consultas analíticas que acessam poucas colunas, uma vez que é possível delimitar os arquivos que o processador de consultas deve varrer. Além

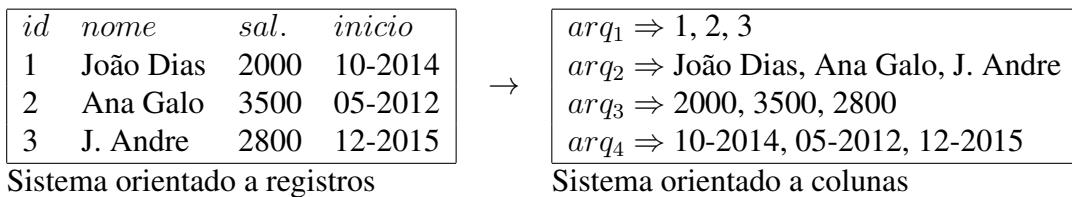


Figura 1. Orientação à registros e à colunas

disso, é possível representar de maneira mais eficiente as colunas que aceitam valores nulos, sem precisar recorrer a mapas de bits indicando campos nulos.

Outro ponto que merece destaque com relação aos bancos orientados a colunas é a sua capacidade de compressão de dados. Nota-se que com essa organização os arquivos passam a ser formados por valores que pertencem ao mesmo domínio e tipo de dados. Isso reduz a entropia da informação, e os algoritmos de compressão podem se beneficiar disso para obter uma taxa de compressão superior.

Caso uma coluna possua baixa cardinalidade (poucos valores distintos), um método de compressão bastante eficaz é a substituição do valor por um código que referencia uma entrada em um dicionário. Já para a compressão de colunas de texto de alta cardinalidade, os métodos que exploram padrões dentro do texto, como LZ (Ziv e Lempel, 1978) e BWT (Burrows e Wheeler, 1994), são mais adequados.

Outro método que encontra grande utilidade na compressão de dados textuais de alta cardinalidade é conhecido como PPM (*Prediction by Partial Matching*) (Moffat, 1990). As taxas de compressão obtidas pelas diversas variações do PPM concorrem com os resultados obtidos pelos demais compressores. O que impede o seu uso mais disseminado é o elevado tempo de processamento.

As estatísticas de desempenho dos compressores PPM são oriundas de testes realizados sobre *benchmarks* conhecidos na área de compressão de dados, como o corpus CALGARY (Council, 2008), formado por arquivos de formatos variados, como textos, imagens e códigos em linguagens de programação. No entanto, bancos de dados orientados a colunas possuem características bem distintas no que diz respeito aos padrões que podem ser encontrados.

Assim sendo, o objetivo desse artigo é investigar o comportamento do PPM na compressão de dados textuais de alta cardinalidade e analisar se seu uso é viável para arquivos orientados a colunas. A avaliação envolve a análise do tempo de execução e a taxa de compressão do PPM, comparando os resultados àqueles que são obtidos pelo LZ e BWT.

O artigo está estruturado da seguinte forma: a seção 2 apresenta resumidamente algumas das principais estratégias de compressão de dados textuais propostas na literatura. Na seção 3 o método de compressão PPM é apresentado, afim de explicar porque esse método parece especialmente adequado para dados orientados a colunas. A seção 4 apresenta experimentos que foram feitos comparando o desempenho de diversos algoritmos de compressão em cenários compostos por arquivos orientados a colunas e arquivos convencionais pertencentes ao corpus *Calgary*. A seção 5 traz as considerações finais.

2. Algoritmos de Compressão de dados

A compressão de dados textuais sem perda recebeu muita atenção da comunidade científica em décadas passadas. Uma das primeiras ideias exploradas foram as chamadas técnicas de codificação estatística, como a codificação de Huffman (Huffman et al., 1952) e a codificação aritmética (Witten et al., 1987). Nos dois casos a compressão é obtida representando os caracteres (ou símbolos) mais frequentes do arquivo usando menos bits.

O código de Huffman emprega uma árvore binária que mapeia símbolos como cadeias de bits. Por sua vez a codificação aritmética emprega uma tabela que guarda a frequências de ocorrências dos símbolos já processados. Sabendo essas frequências, pode-se calcular a probabilidade de ocorrência de qualquer símbolo. Essa probabilidade é então codificada como um valor binário de ponto fixo. Os codificadores de Huffman e aritméticos podem ser estáticos, quando usam árvores/tabelas de frequência pré-determinadas, ou dinâmicos, quando as árvores/tabelas são construídas à medida que a compressão ocorre.

Mais tarde surgiram técnicas de compressão que passaram a levar em consideração não apenas a frequência dos símbolos, mas o fato de que muitos símbolos costumam aparecer juntos. As técnicas que exploram essa característica são conhecidas como baseadas em dicionário. Os algoritmos dessa categoria mais usados hoje em dia derivam de uma ideia proposta por Ziv e Lempel (1978), e são referenciados pelas iniciais de seus autores (LZ). De modo geral, a sequência de símbolos já processada do arquivo de entrada forma o dicionário. Uma sequência de símbolos a codificar é representada através de um índice de deslocamento e uma largura. O índice indica um ponto no arquivo de entrada onde essa sequência já foi encontrada. A largura determina quantos símbolos a partir desse índice são equivalentes aos símbolos que se deseja comprimir. Essas duas informações são representadas através de um codificador estatístico, sendo o código de Huffman mais comumente utilizado. Essa ideia levou à especificação de um padrão para codificação chamado DEFLATE (Deutsch, 1996) e serviu de base para a criação dos compressores de dados mais usados comercialmente, como gzip e lzip.

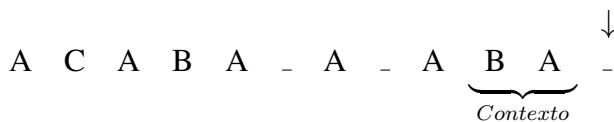
Outra técnica que se mostrou particularmente interessante para a compressão de texto foi proposta por Burrows e Wheeler (1994). Seu nome, BWT, é um acrônimo que remete aos nomes dos autores (*Burrows Wheeler Transform*). O passo inicial do algoritmo gera todas as permutações que se obtém ao rotacionar o texto a comprimir um símbolo de cada vez. Essas permutações são armazenadas em uma matriz onde as linhas são ordenadas. No próximo passo todas as colunas com exceção da última são descartadas. É possível reconstruir o texto original usando apenas essa última coluna e um índice que localiza a linha da matriz onde o texto original estaria armazenado. Esse método parte da constatação de que alguns símbolos são normalmente precedidos por determinados símbolos. Caso isso ocorra, a última coluna da matriz será composta por muitos símbolos repetidos. Isso abre espaço para a aplicação de uma técnica chamada MTF (*Move to Front*) que visa transformar essa saída em outra composta por valores de 0 a 255 com a predominância de valores baixos. O último passo (que é onde a compressão realmente ocorre) envolve codificar essa saída composta por valores numéricos usando algum codificador estatístico como *Huffman* ou codificação aritmética.

Também existem trabalhos voltados à bancos de dados orientados a colunas que exploram colunas que possuem determinadas características (Abadi et al., 2009). Por

exemplo, quando é comum que os valores sejam compostos por muitos caracteres em branco consecutivos, pode-se empregar técnicas de supressão de nulos, cujo objetivo é remover um símbolo de elevada ocorrência (como o espaço em branco, por exemplo), deixando em seu lugar a sua localização e quantidade (Westmann et al., 2000). Caso mais símbolos costumem aparecer de forma consecutiva, uma técnica simples e que encontra empregabilidade em diversas aplicações é a RLE (*Run Length Encoding*), onde símbolos consecutivos repetidos são substituídos por um par composto pelo símbolo e pelo número de repetições. Essa técnica pode ser útil em sistemas orientados a colunas que guardam os valores ordenados (Abadi et al., 2006). Já o uso de dicionários e vetores de bits (Wu et al., 2002) são indicados para casos em que a quantidade de valores distintos é baixa (baixa cardinalidade). De modo geral, esses trabalhos tem objetivos ortogonais aos propostos neste artigo, cujo foco é em dados textuais de alta cardinalidade e que não necessariamente estejam ordenados.

3. PPM

O método de compressão PPM codifica um símbolo de cada vez. Para gerar um código é levado em consideração o contexto, que são os símbolos que precedem o símbolo a ser codificado. Para compreender o funcionamento do PPM, considere o texto a comprimir indicado abaixo. A seta indica o próximo símbolo a ser codificado, e as chaves indicam o contexto a ser analisado.



Dado o símbolo a ser codificado ('-'), a codificação é determinada pela probabilidade de ocorrência desse símbolo dado o contexto que o precede. Para realizar esse cálculo é necessário analisar quais símbolos já ocorreram no passado quando esse contexto foi encontrado, e quais são as frequências de ocorrência desses símbolos. Quanto maior a frequência, maior é a probabilidade. A probabilidade pode ser computada usando codificação aritmética, de modo que símbolos mais prováveis gerem menos bits durante a codificação.

Pela Tabela 1 é possível observar quais símbolos ocorreram até então (e suas frequências) para cada ordem do contexto atual. Por exemplo, para o contexto de maior ordem ('BA') o histórico mostra que apenas um símbolo ocorreu ('-'), tendo ocorrido uma vez. Para cada contexto um símbolo especial é reservado, chamado de escape (ESC). A frequência desse símbolo depende da implementação do PPM. A implementação clássica considera que a frequência é equivalente ao número de símbolos distintos que já ocorreram naquele contexto (Moffat, 1990). O propósito desse símbolo especial será descrito mais adiante.

No caso em questão, o símbolo '-' tem uma probabilidade de ocorrência equivalente a 50% após o contexto 'BA'. Esse percentual é traduzido em um código binário de ponto fixo através de codificação aritmética. Em seguida a tabela de frequência dos contextos é atualizada com o símbolo recém processado e o codificador avança para processar o próximo símbolo.

Tabela 1. Contextos de ordens de zero a dois e seus respectivos símbolos/frequências

| Ordem | Contexto | Símbolo (Frequência) |
|-------|----------|---------------------------------|
| 2 | B A | ESC(1), _-(1) |
| 1 | A | ESC(3), B(2), C(1), _-(2) |
| 0 | | ESC(4), A(5), B(2), C(1), _-(2) |

Caso o símbolo a codificar fosse 'C' em vez de '_' (supondo o texto 'ACABA_A_ABAC'), observa-se que em nenhum momento no passado esse símbolo foi encontrado depois do contexto 'BA'. Nesse caso deve ser codificado o símbolo de escape, com probabilidade de 50%. Esse símbolo sinaliza que o contexto deve ser reduzido (de 'BA' para 'A') e a busca feita novamente. Dessa vez, três símbolo ocorreram após 'A'. Um deles é aquele que se deseja codificar, com probabilidade de 12,5% (uma ocorrência dentre as oito existentes). A existência de probabilidades iguais (ex. a probabilidade de aparecer 'B' ou '_' depois de 'A') é resolvida pela codificação aritmética através da divisão da escala de probabilidades em intervalos.

O pseudo-código do Algoritmo 1 mostra como o contexto diminui de tamanho a medida que a busca avança. No pior dos casos o contexto é reduzido à ordem 0 (zero). Nesse caso leva-se em consideração a frequência total dos símbolos, independente de onde eles apareceram. Todos os símbolo possíveis são contemplados nessa lista, então a busca sempre será bem sucedida nesse nível. A descompressão segue o caminho inverso. Símbolos decodificados alimentam o contexto e servem de indício para a decodificação do próximo símbolo.

Algoritmo 1: CODIFICAÇÃO USANDO PPM

```

Entrada: simbolos_lidos, simbolo_a_codificar
1 início
2   contexto ← ultimos n simbolos_lidos
3   para cada ordem de n a 0 faz
4     no ← busca_simbolo(contexto, simbolo_a_codificar)
5     se no não for nulo então
6       codifica_simbolo(no)
7       retorna
8     fim
9     senão
10      codifica_escape()
11      encurta_contexto()
12    fim
13  fim
14 fim

```

O algoritmo original e muitas de suas variações utilizam um tamanho máximo de contexto. Experimentos indicam que melhores taxas de compressão são obtidas ao utilizar contextos cujo tamanho máximo (n) está compreendido no intervalo de três a sete (Moffat, 1990). Também existem variações que não limitam o tamanho do contexto (Cleary e Teahan, 1997). No entanto, seu consumo de memória é elevado, apesar da preocupação

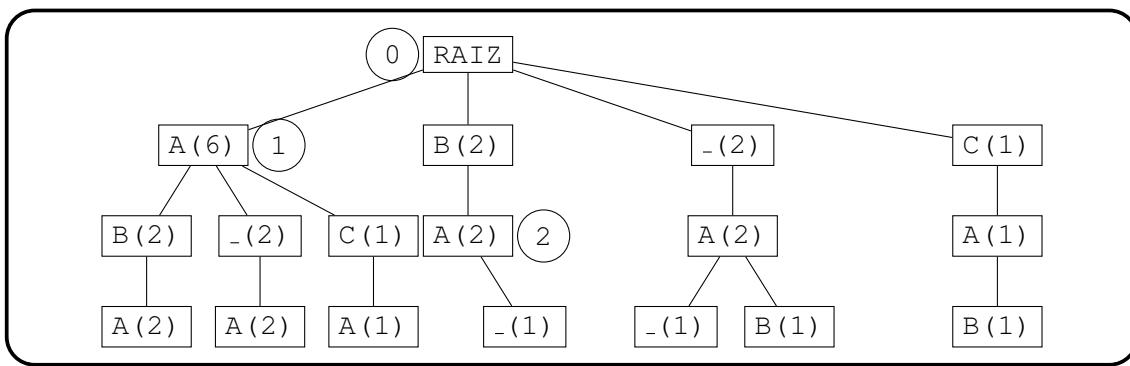


Figura 2. Árvore de contexto (ordem máxima = 2)

no uso de estruturas de dados que minimizem esse custo.

A Figura 2 mostra a árvore de contextos que seria gerada para o texto usado como exemplo, considerando um tamanho máximo de contexto igual a 2 e um universo de símbolos composto por 'A', 'B', 'C' e '_'. O contexto de cada nó é identificado pela concatenação do símbolo do nó atual e dos símbolos dos nós ascendentes. Para cada nó é também armazenada a frequência de ocorrência do contexto correspondente. Dentro de um nó os filhos são ordenados pela frequência. Essa organização visa obter menores tempos na busca de um símbolo (linha 4 do pseudo-código). Caso esse símbolo seja muito frequente, poucos nós deverão ser visitados até que ele seja encontrado.

O nível zero é reservado para o contexto vazio. Todos os símbolos possíveis aparecem como filhos do raiz com uma frequência não nula, para que sua probabilidade de ocorrência seja superior a zero. Isso garante que símbolos que nunca sucederam um contexto específico possam ser codificados quando o contexto for encurtado até ficar vazio.

Os círculos indicam os nós que fazem parte do contexto atual, ou seja, os nós que identificam os símbolos presentes em 'BA', em cada uma das ordens, de zero a dois. A operação que encura contexto (linha 11 no pseudo-código) pode ser vista como a navegação de um nó de ordem maior para um nó de ordem menor.

A taxa de compressão dos algoritmos PPM depende fortemente da existência de padrões de repetição nos símbolos a processar. Isso é bastante comum em textos, onde as mesmas palavras (ou compostas pelo mesmo radical) costumam aparecer com frequência. Isso leva à geração de árvores em que cada pai possua poucos filhos muito frequentes e muitos filhos pouco frequentes. Nesses casos, as sequências de caracteres que costumam aparecer juntas são codificadas com poucos bits, uma vez que a probabilidade de ocorrência dos caracteres dentro dessas palavras comuns será alta.

Considerando uma organização de arquivos orientado a coluna, em que os valores de cada coluna são armazenados de forma consecutiva, a expectativa é que sejam encontrados ainda mais padrões do que o que se costuma encontrar em arquivos de texto convencionais. A próxima seção investiga como a natureza dos dados encontrados em arquivos orientados a colunas se relaciona com o PPM e com outros compressores.

4. Experimentos

Os experimentos desta seção mostram como diversos compressores de texto se comportam ao comprimir dados orientados a colunas. Os compressores testados são o PPM, codificação aritmética, BWT e LZ. Todos foram implementados em C. Os dois primeiros foram criados como parte deste trabalho. Para os dois últimos foram utilizados os compressores BZIP2 e GZIP, respectivamente. Nenhuma *flag* de otimização foi utilizada na compilação/execução dos códigos-fontes.

Os dados das colunas foram gerados a partir do TPC-H, um conhecido *benchmark* usado para avaliar a performance de processamento de transações de bancos de dados (Council, 2008). O modelo de dados do TPC-H é composto por oito tabelas (PART, SUPPLIER, PARTSUPP, CUSTOMER, NATION, LINEITEM, REGION, ORDERS). O gerador de dados foi configurado com um fator de escala igual a 1, o que resultou em um volume de dados de aproximadamente 1 GB.

Após a geração dos dados, as tabelas foram segmentadas de modo a se aproximar da organização física de arquivos empregada em SGBDS orientados a colunas. Para isso, uma tabela com x colunas foi dividida em x arquivos distintos. Em cada arquivo os valores da coluna respectiva foram adicionados consecutivamente, separados um do outro por um símbolo de uso reservado.

Foram selecionadas para compressão apenas colunas que armazenassesem tipos de dados textuais e tivessem alta cardinalidade. Diversas colunas que satisfaziam os critérios foram avaliadas. De modo geral, em todas elas o resultado foi semelhante. Desse modo, foi escolhida a coluna COMMENT da tabela CUSTOMER como referência.

A primeira análise é voltada exclusivamente ao método de compressão PPM. A intensão é descobrir o melhor tamanho de contexto para o domínio de dados escolhido de modo a obter melhores taxas de compressão sem que isso acarrete em perdas significativas de desempenho. A relação entre esses dois fatores (compressão x desempenho) se dá basicamente pelo tamanho da árvore gerada. Contextos curtos geram árvores menores. Assim, gasta-se menos tempo na manutenção da árvore. Por outro lado, a probabilidade de ocorrência de um símbolo qualquer tende a ser menor, o que diminui a taxa de compressão.

A Figura 3 apresenta resultados empíricos relacionando o tamanho do contexto aos fatores desempenho (gráfico da esquerda) e taxa de compressão (gráfico da direita). O desempenho é medido como o tempo necessário em milissegundos para realizar a compressão. A taxa de compressão é medida em bits por código(bpc), que indicam quantos bits são necessários para compactar cada byte do arquivo de entrada. Foram testadas diversas versões do PPM, variando o tamanho máximo do contexto de dois (PPM-2) até sete (PPM-7). Os gráficos permitem ver como os resultados variam conforme parcelas maiores do arquivo COMMENT são processadas.

Como pode-se ver, o PPM-2 obteve o pior desempenho nos dois fatores analisados. Apesar de pouco tempo ser gasto na geração da árvore, muito tempo é investido na busca de nós a partir de um pai. Como os nós filhos são ordenados pela frequência, a busca por nós com baixa frequência provoca um acesso a um maior número de filhos até que se encontre o nó correto. Além disso, como o arquivo comprimido é maior, perde-se mais tempo em operações de gravação do arquivo de saída.

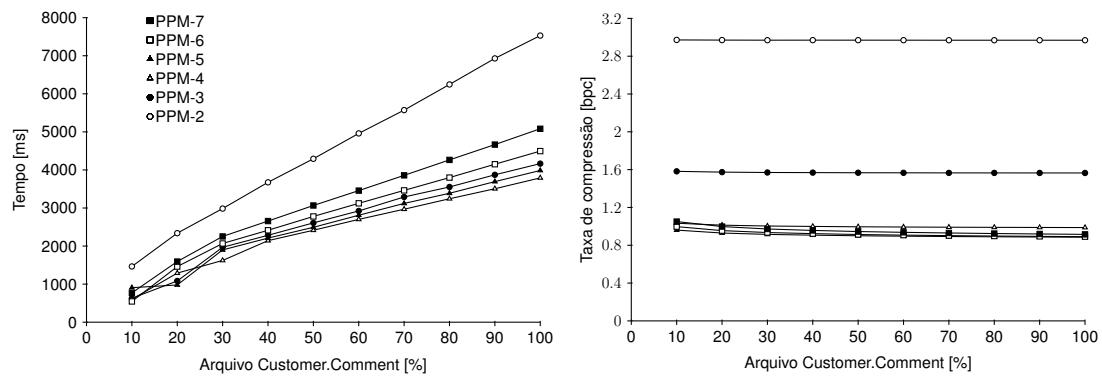


Figura 3. Diferentes versões do PPM variando o tamanho máximo do contexto

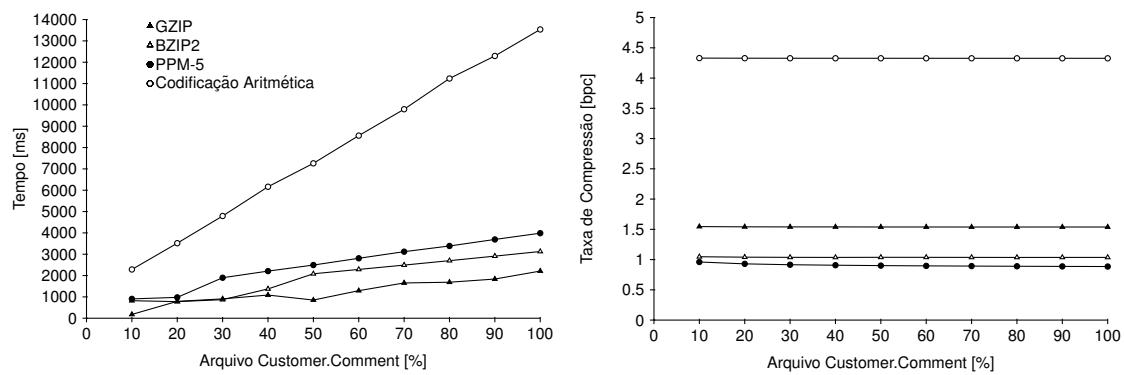


Figura 4. Algoritmos de compressão processando um arquivo orientado a colunas

O PPM-7 também gerou resultados insatisfatórios quanto ao tempo de processamento. Isso ocorre em boa parte porque a manutenção da árvore requer muito trabalho. Por outro lado, o PPM-7 obteve uma boa taxa de compressão. No entanto, a taxa é semelhante aos resultados obtidos por versões do algoritmo que usaram tamanhos máximos de contexto menores. Analisando os dois fatores em conjunto, percebe-se que os PPM-4 e PPM-5 apresentam uma boa relação custo-desempenho, sendo que o PPM-4 é levemente superior no quesito desempenho enquanto o PPM-5 é melhor na taxa de compressão. Como o objetivo é atingir boas taxas de compressão sem perdas significativas de desempenho, a versão PPM-5 foi utilizada no demais experimentos.

Os próximos gráficos(Figura 4) comparam o desempenho e a taxa de compressão de todos os algoritmos avaliados. Novamente a medição foi feita considerando parcelas do arquivo COMMENT. Os resultados mostram que a compressão aritmética pura perde nos dois fatores. Os outros três algoritmos apresentam um custo benefício semelhante, sendo que o GZIP apresenta o menor tempo de execução enquanto o PPM-5 apresenta a maior taxa de compressão. Caso a intenção seja otimizar a ocupação de espaço em disco, a alternativa que implementa o PPM seria preferível.

Para finalizar, os gráficos da Figura 5 incluem na comparação com a coluna COMMENT o corpus CALGARY. Esse corpus possui uma coleção de arquivos de variados formatos e tamanhos, e é bastante utilizado como *benchmark* de compressão de dados (Arnold e Bell, 1997). A tabela CUSTOMER, de onde foi extraída a coluna COMMENT, também foi adi-

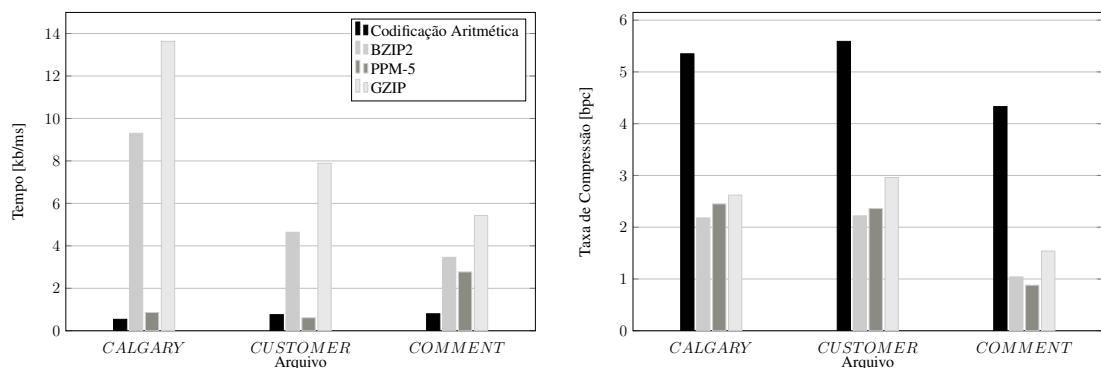


Figura 5. Algoritmos de compressão processando arquivos variados

cionada. Como essa tabela é orientada a registros, cabe fazer uma análise referente ao comportamento dos algoritmos de acordo com a organização física dos dados. Os arquivos CALGARY, CUSTOMER e COMMENT ocupam respectivamente 3.2, 23.9 e 10.8 mbytes.

O gráfico da esquerda exibe a velocidade de processamento, medida em kbytes processados por milissegundo. Aqui pode-se ver que BZIP2 e GZIP são visivelmente mais rápidos do que o PPM e a codificação aritmética na compressão de arquivos convencionais ou da tabela orientada a registros. No entanto, essa relação de desempenho é menos impactante na compressão do arquivo orientado a colunas. Além disso, a velocidade do GZIP e BZIP2 diminui na compressão do arquivo COMMENT, enquanto a velocidade do PPM aumenta. Isso mostra que a proximidade de valores de um mesmo domínio impulsiona o PPM e causa um efeito contrário em seus principais concorrentes.

Já o gráfico da direita exibe a taxa de compressão. Todos os compressores obtiveram melhores resultados no arquivo orientado a colunas. Isso demonstra que a redundância desse tipo de arquivo é bem explorada pelos algoritmos. O que vale a pena destacar aqui é a distinção que existe na compressão do CALGARY e CUSTOMER em comparação à COMMENT. Nos dois primeiros, BZIP2 e GZIP foram superiores ao PPM. Já no arquivo COMMENT essa relação se inverteu. Esse resultado sugere que métodos baseados em contexto como o PPM são mais eficazes na compressão de informações textuais cujo universo de valores aceitos pertence a um domínio de dados mais restrito.

5. Conclusões

Arquivos orientados a colunas são realmente bem explorados por compressores de dados baseados em padrões. Os experimentos realizados mostraram que as taxas de compressão são maiores quando se lida com dados bem comportados cujos valores pertencem a um domínio de dados bem definido, uma vez que a entropia tende a ser menor.

Outro ponto importante levantado nos experimentos foi a descoberta de que o método de compressão PPM se mostra particularmente viável para esse tipo de dados, apresentando taxas de compressão superiores aos métodos concorrentes e um tempo de processamento não muito superior. Aqui cabe ressaltar que também foram realizados experimentos medindo tempo de execução na descompressão. Esses resultados foram omitidos por serem análogos aos resultados obtidos na compressão, posicionando o PPM como método de compressão com desempenho razoavelmente competitivo.

Os resultados obtidos servem de motivação para a investigação de formas de melhorar o desempenho do PPM. Nessa linha de pesquisa, um dos pontos a serem explorados surgiu de uma constatação feita durante os experimentos. Conforme a Figura 3 indica, a taxa de compressão do PPM permanece constante ao longo do processamento do arquivo orientado a colunas. Isso sugere que a árvore de contextos existente após o processamento do trecho inicial do arquivo apresenta uma relação de probabilidades similar à árvore de contextos existente após o processamento de todo o arquivo. Assim, a ideia a investigar é a interrupção na manutenção da árvore de contexto quando alguns critérios forem atingidos, na expectativa de que a árvore existente seja um modelo de predição bom o suficiente para a codificação dos próximos símbolos.

Referências

- Abadi, D., Madden, S., e Ferreira, M. (2006). Integrating compression and execution in column-oriented database systems. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 671–682. ACM.
- Abadi, D. J., Boncz, P. A., e Harizopoulos, S. (2009). Column-oriented database systems. *Proceedings of the VLDB Endowment*, 2(2):1664–1665.
- Arnold, R. e Bell, T. (1997). A corpus for the evaluation of lossless compression algorithms. In *Data Compression Conference, 1997. DCC'97. Proceedings*, pages 201–210. IEEE.
- Burrows, M. e Wheeler, D. J. (1994). A block-sorting lossless data compression algorithm (relatório técnico).
- Cleary, J. G. e Teahan, W. J. (1997). Unbounded length contexts for ppm. *The Computer Journal*, 40(2 and 3):67–75.
- Council, T. P. P. (2008). Tpc-h benchmark specification. [Online; acessado em 19 de janeiro de 2016].
- Deutsch, L. P. (1996). Deflate compressed data format specification version 1.3.
- Han, J., Haihong, E., Le, G., e Du, J. (2011). Survey on nosql database. In *Pervasive computing and applications (ICPCA), 2011 6th international conference on*, pages 363–366. IEEE.
- Huffman, D. A. et al. (1952). A method for the construction of minimum redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101.
- Moffat, A. (1990). Implementing the ppm data compression scheme. *Communications, IEEE Transactions on*, 38(11):1917–1921.
- Westmann, T., Kossmann, D., Helmer, S., e Moerkotte, G. (2000). The implementation and performance of compressed databases. *ACM Sigmod Record*, 29(3):55–67.
- Witten, I. H., Neal, R. M., e Cleary, J. G. (1987). Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540.
- Wu, K., Otoo, E. J., e Shoshani, A. (2002). Compressing bitmap indexes for faster search operations. In *Scientific and Statistical Database Management, 2002. Proceedings. 14th International Conference on*, pages 99–108. IEEE.
- Ziv, J. e Lempel, A. (1978). Compression of individual sequences via variable-rate coding. *Information Theory, IEEE Transactions on*, 24(5):530–536.

aper:152931_1

Estratégias para importação de grandes volumes de dados para um servidor PostgreSQL *

**Vanessa Barbosa Rolim¹, Marilia Ribeiro da Silva¹, Vilmar Schmelzer¹
Fernando José Braz¹, Eduardo da Silva¹**

¹Fábrica de Software, Instituto Federal Catarinense – Câmpus Araquari

{nessabrolim, marilia.ifc, vilmarssss}@gmail.com

{fernando.braz, eduardo}@ifc-araquari.edu.br

Abstract. Among the projects in development on Fabrica de Software environment at the Catarinense Federal Institute, one is about a traffic management support system. In this project, the import of an external, too large, and unstructured file data to an internal SQL database is required. The use of a framework Django to import this data obtained a low performance result. Then, new import strategies were desirable. Thus, this work deals with the optimization of data import strategies used to solve this problem, by presenting the proposed solutions and comparing their performance results.

Resumo. Dentre os projetos desenvolvidos no ambiente da Fábrica de Software do Instituto Federal Catarinense, um trata de um sistema de gestão de trânsito. No âmbito desse projeto, é necessário importar uma base de dados externa muito grande e não estruturada para uma base interna em SQL. A utilização de um framework Django para a importação dos dados resultou em baixo desempenho, iniciando a busca por novas estratégias de importação. Assim, esse trabalho trata da otimização das estratégias de importação de dados utilizadas para a resolução desse problema, apresentando as soluções propostas e comparando o desempenho dos resultados.

1. Introdução

A Fábrica de Software, vinculada ao Núcleo de Operacionalização e Desenvolvimento de Sistemas de Informação (NODES) do Instituto Federal Catarinense, abriga, entre outros, um projeto de desenvolvimento de software em parceria com o Departamento de Trânsito de Joinville/SC (Detrans) [Mota and et al. 2014]. Um de seus objetivos trata da inconsistência e redundância de dados, visto que a base de dados utilizada atualmente pelo Detrans fica contida em uma série de arquivos de texto.

Esses arquivos se referem às características de veículos (cor, espécie e categoria), características de infração (tipo de infração, lei à qual se refere) e ao registro dos veículos em si. O conteúdo possui codificações diferentes dentro do mesmo arquivo, sendo o UTF-8¹ a codificação predominante. Cada arquivo possui em média 50 registros, com exceção daquele que contém os dados dos veículos, doravante chamado arquivo1. Esse arquivo,

*Projeto de pesquisa parcialmente apoiado pelo CNPq (488004/2013-6) e do edital MEC/SETEC 94/2013

¹Mais informações em <http://www.rfc-editor.org/info/rfc3629>.

no dia 20 de maio de 2015, continha cerca de 4,5 milhões de registros, ocupando 736,8 MB de espaço em disco.

Para manter a integridade e o desempenho do sistema, os dados devem ser importados para uma nova base no servidor de banco de dados. Os primeiros testes de importação realizados estimaram pelo menos uma semana para o *upload* dos dados. Esse resultado é insatisfatório para o cenário atual e projeção futura, pois é uma tarefa rotineira.

Diante disso, este artigo trata das soluções encontradas para a otimização do tempo de *upload* dos arquivos e da importação dos dados para um servidor de banco de dados. Após o término do *upload* dos registros, esse tempo passa a ser, aproximadamente, de duas horas com a otimização, que considerou fatores como a modelagem do banco de dados, a leitura de arquivos e a divisão dos dados dos arquivos por conteúdo.

Este trabalho está organizado da seguinte forma: Seção 2 descreve o cenário do projeto. Seção 3 apresenta a proposta e seus experimentos. Seção 4 apresentação dos resultados obtidos. Seção 5 conclui o artigo e apresenta perspectivas de trabalhos futuros.

2. Cenário

O arquivo1 é um dos arquivos que contém dados sobre veículos do estado de Santa Catarina, disponibilizado pelo Detrans. Dado que o Detrans está passando por um processo de implantação de novas tecnologias, fez-se necessária a importação dos dados deste, e de outros arquivos, para um servidor de banco de dados. Esses dados serão integrados com o sistema web de Gestão de Trânsito da cidade de Joinville que está sendo desenvolvido pela Fábrica de Software.

Para a construção do projeto, são utilizados um servidor de aplicação Django 1.8, com Python 2.7, e Postgres 9.3. Para o ambiente web utiliza-se um servidor virtual sobre VMWare 5 ESXI, que roda Debian 8.0 GNU/Linux, com 4 processadores Intel Xeon de 2.13 GHz e 2 GB de memória RAM.

O arquivo1 foi obtido através de acesso remoto utilizando o protocolo de transferência de arquivos FTP. Além desse, outros arquivos também foram transferidos para a execução do projeto. Porém, esse artigo trata dos problemas referente ao *upload* dos dados do arquivo1 para uma base de dados relacional.

Visto que o arquivo1 é um arquivo de texto, a Figura 1 apresenta a sua estrutura. Pode-se dizer que sua estrutura se assemelha aos arquivos do tipo csv, porém não possui caracteres delimitadores. No arquivo1 cada atributo de veículo possui um tamanho fixo de caracteres e, como pode ser observado na Figura 1, encontram-se armazenados de forma sequencial. Então, cada linha do arquivo deve conter o somatório de caracteres de todos os atributos, ou seja, 142 caracteres.

| placa | marca | cor | tip | esp | cat | municíp | renavam | dt fab | dt mod |
|----------|-----------|-----|-----|-----|-----|---------|--------------|--------|--------|
| cpf/cnpj | | | | | | | proprietário | | |
| pass | nº chassi | | | | | | nº motor | | |
| 102 | | | | | | 122 | | 142 | 99 |

Figura 1. Formato do arquivo1 por atributo de veículo em relação a quantidade de caractere.

Como os registros foram inseridos em um banco de dados relacional, estudou-se as características dos atributos a fim de definir um identificador único para veículo. Assim, a coluna placa não pode ser utilizada como chave primária, pois se repete indefinidas vezes no arquivo1 que foi apontado com várias inconsistências, entre elas a codificação e a duplicidade parcial de registros. Optou-se, então, pela utilização da coluna chassi como a chave primária de veículo. As outras características tornaram-se chaves estrangeiras para o modelo relacional.

A Figura 2 ilustra a modelagem resumida do banco de dados. Como é possível observar, a entidade veículo é subordinada às outras, com exceção de proprietário, pois depende de uma série de chaves estrangeiras para que possa existir[Date 2004]. Além disso, um veículo pode ter uma série de proprietários ao longo do tempo, assim criou-se uma associação entre essas entidades.

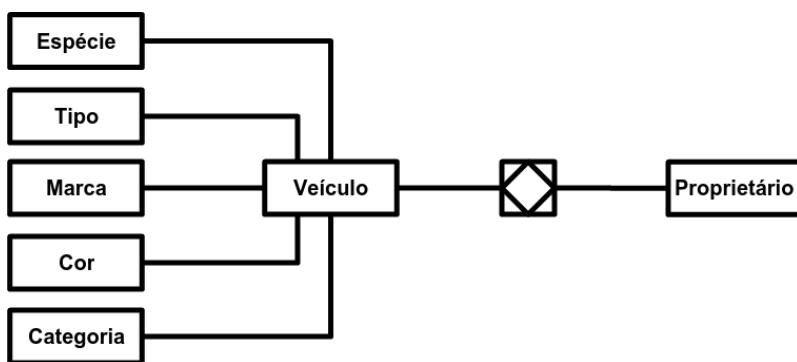


Figura 2. Modelagem resumida (simplificada) do banco de dados.

Após estudo do ambiente, foram realizadas diversas experimentações em baixa e larga escala, estudos matemáticos, modelagem do sistema, pesquisa de técnicas de conversão de arquivos e bibliográfica.

3. Soluções Propostas

A seguir são apresentadas duas propostas de inserção dos registros na nova base de dados. A primeira proposta possui uma abordagem convencional utilizada pelo framework Django e a segunda possui um caráter de otimização para a realidade da aplicação em questão.

Em ambas as propostas foram realizadas tarefas de seleção e pré-processamento de dados, aplicando-se um algoritmo Python, manipulado pelo framework Django, para a limpeza dos dados e definição dos conjuntos de dados consistentes e inconsistentes. Em outras palavras, esse algoritmo fez a leitura do arquivo1 e classificou em dados consistentes aqueles registros que seguiam a codificação UTF-8 e que não possuíam duplicatas. Os demais registros foram mantidos no arquivo de inconsistências para serem tratados numa nova etapa do projeto.

3.1. Proposta 1: inserção convencional dos registros

A primeira proposta consiste na importação dos dados consistentes do arquivo1 utilizando o método clássico de persistência de dados executadas pelo framework Django. Como mencionado, os dados passaram por um processo de seleção e pré-processamento,

como ilustrado pela Figura 3. Logo, à medida que as linhas do arquivo eram lidas classificadas como registros consistentes, chamava-se o método padrão de inserção em banco de dados utilizados pelo framework Django. Esses métodos são generalizados, assim, a consistência dos dados se dá através de uma busca completa por identificadores únicos de proprietário a cada inserção de veículo. Caso o proprietário não exista, um novo registro de proprietário é inserido, e posteriormente, sua chave primária é inserida na tabela associativa. Quando o CPF ou CNPJ já estiver cadastrado na tabela de proprietários, faz-se apenas sua associação com o veículo.

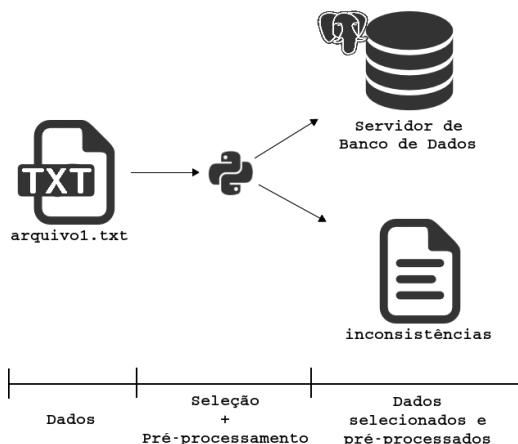


Figura 3. Processo de importação convencional da primeira proposta.

Foram realizados testes através da importação da base completa. Os resultados, apresentados na Seção 4, demonstraram que a adoção de uma nova proposta de importação era necessária.

3.2. Proposta 2: inserção otimizada dos registros

A segunda proposta trata da separação dos dados, através de uma função Python, em dois arquivos (`veiculos.csv` e `proprietarios.csv`) e sua posterior inserção no banco de dados, Através de outra função Python, conforme ilustrado na Figura 4.

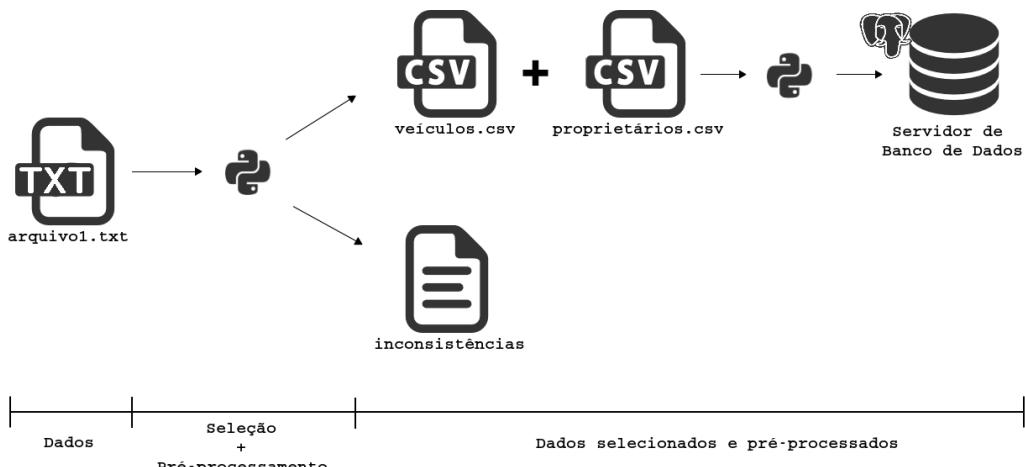


Figura 4. Processo de importação: separação em arquivos especialistas.

Para garantir a integridade dos dados, proprietarios.csv é percorrido em busca da chave primária (chassi ou CPF/CNPJ): caso repetições sejam encontradas para a mesma chave, apenas a primeira ocorrência é mantida.

Os veículos são inseridos através de um código SQL que não considera as características como chaves estrangeiras, mas como texto. Dessa forma, quando as outras tabelas são populadas, as chaves primárias são as mesmas que nos arquivos de texto puro, e a ligação pela chave estrangeira fica garantida. A seguir, a entidade associativa entre veículos e proprietários é construída, utilizando-se o chassi e o CPF/CNPJ contidos no arquivo veiculos.csv.

O código abaixo (Listing 1), trata do processo de importação de veículos para o banco de dados. A função foi resumida para aumentar a clareza em sua leitura.

Listing 1. Código que implementa função de inserção de registros no servidor de banco de dados.

```
def importa_veiculos(veiculos):
    erros_importa_veiculo = []
    cur = connection.cursor()

    try:
        cur.executemany(insert_veiculo, veiculos)
        connection.commit()
    except Exception as e:
        connection.rollback()

    for insert_item in veiculos:
        try:
            cur.executemany(insert_veiculo, [insert_item])
            connection.commit()
        except Exception as ex:
            connection.rollback()

        erros_importa_veiculo.append(
            {'erro': 'insert_veiculo', 'ex': '%s;%s' %
             (type(ex), ex), 'item': insert_item})
    cur.close()

    return erros_importa_veiculo
```

A função importa_veiculos recebe uma lista de veículos, provindos do arquivo veiculos.csv. Após o início de uma nova conexão com o banco de dados, ocorre a tentativa de efetivar o salvamento das informações. Caso ocorra algum erro, ele é inserido em uma lista de erros. Por fim, a função retorna uma lista de erros.

Como pode ser observado no código abaixo (Listing 2) os proprietários são vinculados aos veículos, através da função vincula_veiculo_proprietario. Essa função recebe uma lista de veículos, e uma vez que a conexão com o banco de dados é concluída, tenta inserir os dados do proprietario e do veiculo como texto puro. Caso ocorra alguma falha

durante a inserção, o erro é inserido em uma lista de erros, que é por fim retornada.

Listing 2. Código que implementa a vinculação de veículos aos respectivos proprietários durante a inserção de registros no servidor de banco de dados.

```
def vincula_veiculo_proprietario(veiculo_proprietario_list):
    insert_veiculo_proprietario = '''INSERT INTO
        detransapp_veiculoproprietario(veiculo_id , proprietario_id ,
        data) values (%s,%s ,now())'''

    erros_importa_veiculo_proprietario = []
    cur = connection.cursor()

    try :
        cur.executemany(insert_veiculo_proprietario ,
            veiculo_proprietario_list)
        connection.commit()
    except Exception as e:
        connection.rollback()

    for insert_item in veiculo_proprietario_list:
        try :
            cur.executemany(insert_veiculo_proprietario ,
                [insert_item])
            connection.commit()
        except Exception as ex:
            connection.rollback()

        erros_importa_veiculo_proprietario.append({
            'erro': 'insert_veiculo_proprietario',
            'ex': '%s;%s' % (type(ex), ex),
            'item': insert_item})
    cur.close()

    return erros_importa_veiculo_proprietario
```

4. Resultados e Discussão

Na primeira proposta, a importação foi executada por cerca de 72 horas no servidor. Durante esse período, o processador utilizou 100% de sua capacidade. Estima-se que a importação do arquivo completo levaria cerca de 170 horas (7 dias). O gráfico da Figura 5 demonstra o tempo necessário para a inserção de veículos utilizando as técnicas descritas. Como é possível observar, a medida que os registros são inseridos, o tempo aumenta de forma exponencial. Esse aumento ocorre devido às consultas feitas pelo banco de dados, sejam elas: busca por chaves primárias de veículos duplicados, busca pelas chaves estrangeiras, inserção de proprietários na respectiva tabela e a consequente consulta por chaves primárias repetidas.

Observa-se que o processo descrito se refere à importação tradicional dos dados,

utilizando-se dos pacotes nativos do framework, ou seja, além das operações tradicionais do banco de dados, conforme descrito por [Schatz et al. 2010], são realizadas operações de alto nível específicas do framework Django. Por definição, um framework é uma ferramenta de uso generalizado, um conjunto de funções genéricas que auxiliam no desenvolvimento [Pree 1995], de tal forma que uma aplicação específica, como esse trabalho, sofre com o baixo desempenho das operações em alto nível realizadas pelo framework.

O gráfico da Figura 5 ilustra, ainda, o processo de inserção de dados de veículos no servidor conforme descrito na segunda proposta. Assim, é possível perceber o ganho em desempenho, uma vez que a inserção de veículos necessita de apenas 2 horas, ou seja, quando a responsabilidade pela consistência dos dados é retirada do framework, e tratada através de código pela equipe, as operações de inserção têm melhor desempenho, devido ao fato de que a busca por chaves duplicadas é desconsiderada.

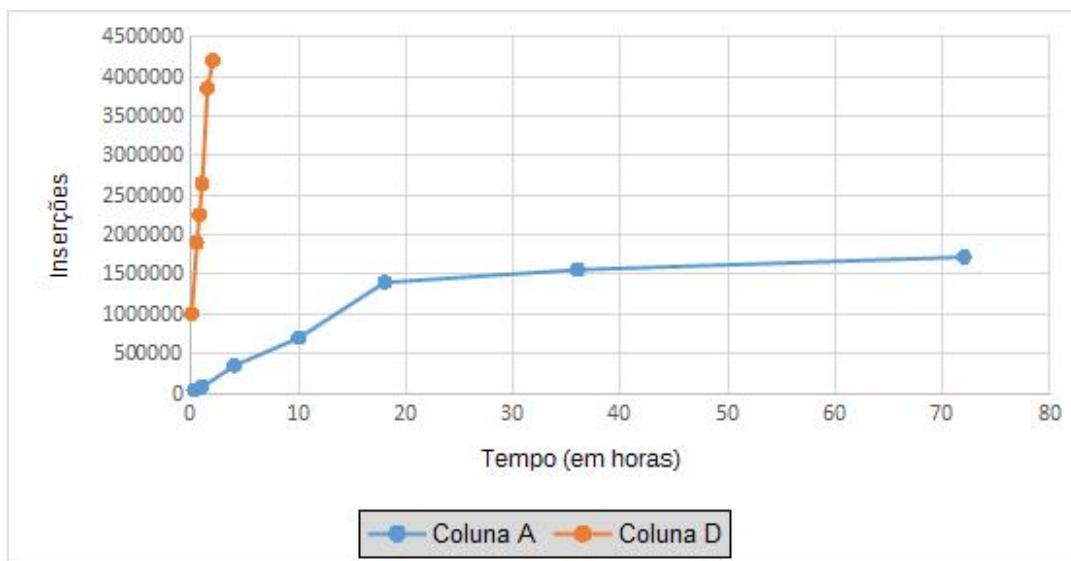


Figura 5. Gráfico de inserção de registros em relação ao tempo da estratégia inicial.

É importante ressaltar que os registros inconsistentes representam uma parcela de aproximadamente 5% dos dados, de tal forma que foram desconsiderados. Assim, eliminou-se o ruído que pode ser uma causa da lentidão do processo inicial de importação. Os dados foram mantidos para que, em uma nova etapa do projeto, sejam restaurados e o banco de dados mantenha a totalidade dos dados iniciais.

5. Considerações Finais

O objetivo principal desse trabalho foi apresentar uma solução de otimização de importação dos dados, oriundos de arquivos de texto, para um servidor de banco de dados relacional. Foram apresentadas duas propostas para a solução do problema. A primeira proposta tratou do inserção dos registros apenas utilizando o suporte oferecido pelo framework Django. A segunda proposta consistiu na separação dos dados do arquivo original, o arquivo1, em outros dois arquivos formatados, veiculos.csv e proprietarios.csv.

Os resultados mostraram que a segunda proposta se mostrou mais eficiente para a solução do problema, uma vez que a inserção de dados foi realizada em tempo menor. Além disso, as buscas por chaves, realizadas pelo banco de dados foram reduzidas, visto que os dados foram separados em novos arquivos específicos.

Com esse trabalho, demonstra-se que a mudança de estratégia pode ser a solução para problemas de baixo desempenho na importação de grandes volumes de dados. O tratamento dos dados e da integridade feito fora do banco de dados resolveu os problemas de desempenho. Sem a perda da generalidade, ao se considerar que a grande maioria dos dados utiliza a codificação UTF-8, aqueles aqueles que não estão nesse grupo foram desconsiderados.

Como parte dos dados devem ser mantidos em uma base de dados nos dispositivos móveis, os objetivos futuros incluem a utilização de índices no processo de popular essa base de dados. Para isso, será utilizado a arquitetura REST para estabelecer a comunicação entre os ambientes web e móvel descrita por [Rolim et al. 2014].

Além disso, pretende-se estudar novas estratégias de otimização utilizando outras tecnologias de persistência de dados, como por exemplo o NoSQL. Ainda, pretende-se verificar a influência da memória cache durante a inserção dos registros.

Referências

- [Date 2004] Date, C. J. (2004). *Introdução a Sistemas de Banco de Dados*. Campus, 8 edition.
- [Mota and et al. 2014] Mota, C. J. and et al. (2014). A experiência do ambiente da Fábrica de Software nas atividades de ensino do curso de Sistemas de Informação do IFC - Campus Araquari. *Anais do XXXIV Congresso da Sociedade Brasileira de Computação – CSBC 2014*, pages 1539 – 1548.
- [Pree 1995] Pree, W. (1995). Design Patterns for Object-Oriented Software Developmen. *Addison Wesley*.
- [Rolim et al. 2014] Rolim, V. B., Silva, M. R. d., Holderbaum, J., and Silva, E. d. (2014). A utilização da arquitetura REST para a comunicação entre diferentes plataformas. *Anais da VII Mostra Nacional de Iniciação Científica e Tecnológica Interdisciplinar - MICTI*.
- [Schatz et al. 2010] Schatz, L. R., Viecelli, E., and Vigolo, V. (2010). PERSISTE: Framework para persistência de dados isolada à regra de negócios. *Congresso Sul Brasileiro de Computação*, 5.

aper:152863_1

Identificação de contatos duplicados em dispositivos móveis utilizando similaridade textual

Rafael F. Machado, Rafael F. Pinheiro, Eliza A. Nunes, Eduardo N. Borges

Centro de Ciências Computacionais – Universidade Federal do Rio Grande (FURG)
Av. Itália, km 8, Campus Carreiros, Rio Grande – RS

{rafaelmachado, rafaelpinheiro, elizanunes, eduardoborges} @furg.br

Abstract. Redundant and often incomplete information substantially reduces the productivity provided by mobile devices. This paper specifies a method to identify duplicate contacts from different data sources, such as e-mail accounts, social networks and those manually set by the user. Using multiple similarity functions, stored records are reorganized in groups of contacts representing the same person or organization. The experiments showed that the proposed method correctly identified up to 76% of duplicated contacts.

Resumo. Informações redundantes e muitas vezes incompletas reduzem consideravelmente a produtividade oferecida pelos dispositivos móveis. Este artigo especifica um método para identificar contatos duplicados coletados de fontes de dados distintas, tais como contas de e-mail, redes sociais e aqueles inseridos manualmente pelo usuário do dispositivo. Utilizando múltiplas funções de similaridade, os registros armazenados são reorganizados em grupos de contatos que representam a mesma pessoa ou organização. Os experimentos realizados mostraram que o método proposto identificou corretamente até 76% dos contatos similares duplicados.

1. Introdução

Nos últimos anos, a Internet e a Web revolucionaram o modo como as pessoas se comunicam. Com a explosão do número de aplicações Web disponíveis, os usuários tendem a acumular diversas contas em diferentes serviços como *e-mail*, redes sociais, *streams* de música e vídeo, lojas virtuais, entre outros. O avanço da tecnologia e a redução do seu custo têm proporcionado o acesso a todos os serviços mencionados de qualquer lugar, a partir de dispositivos móveis como *smartphones* e *tablets*.

Gerenciar informações provenientes de múltiplos serviços ou aplicações é uma tarefa complexa para o usuário. Alguns serviços básicos do dispositivo móvel podem ser prejudicados pela redundância da informação coletada automaticamente por diferentes aplicações. Por exemplo, navegar na lista de contatos com tantas informações repetidas e muitas vezes incompletas reduz consideravelmente a produtividade que o dispositivo móvel pode oferecer.

A Figura 1 apresenta uma porção de uma lista de contatos real composta por dez registros, obtidos de uma ou mais fontes de dados distintas representadas pelos ícones à direita. Algumas informações já estão combinadas de duas ou mais fontes de dados, como é o caso do registro 3. Entretanto, os registros 4, 5, 6 e 8 representam a mesma pessoa e poderiam ser integrados ao registro 3 formando o conjunto D. Ainda poderiam ser integrados os pares de registros A = {1,2} e B = {7,9}, pois representam o mesmo contato. O registro 10 não apresenta duplicatas, portanto deve permanecer isolado.

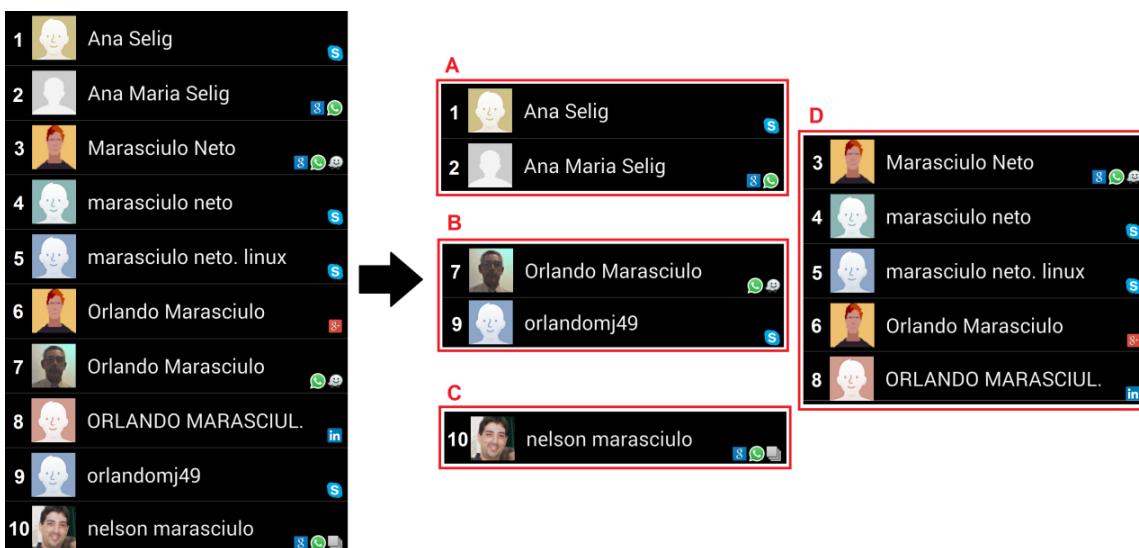


Figura 1. Exemplo de lista de contatos (à esquerda) incluindo registros duplicados e o resultado esperado da deduplicação (à direita).

Sistemas operacionais populares para dispositivos móveis, como iOS [Lecheta 2014] e Android [Ableson 2012], oferecem de forma nativa uma funcionalidade de associação de contatos em que o usuário precisa selecionar os registros que deseja combinar manualmente. Esta tarefa de associação, além de custosa, é armazenada no dispositivo. Se por ventura o usuário perder o dispositivo ou tiver que reinstalar o sistema, os contatos restaurados do backup de sua conta online não estarão associados.

O presente trabalho especifica um método para identificar contatos duplicados coletados de múltiplas fontes de dados que pode ser utilizado como parte da estratégia de associação (integração) automática de contatos. Este método apresenta um avanço significativo no processo de deduplicação proposto originalmente [Pinheiro et. al 2014], pois funções de similaridade são utilizadas no lugar de comparações por igualdade. O método é comparado às estratégias implementadas por um conjunto de aplicativos para gerência de contatos disponíveis gratuitamente. A qualidade do método ainda é avaliada de forma experimental sobre uma base de dados real com quase 2000 registros.

O restante do texto está organizado da seguinte forma. A Seção 2 apresenta um estudo sobre um conjunto de cinco aplicativos para gerência de contatos. Na Seção 3 são revisados trabalhos da literatura científica sobre conceitos fundamentais para o entendimento do trabalho proposto. A Seção 4 especifica o método proposto para deduplicação de contatos. O protótipo desenvolvido e os resultados da avaliação experimental são discutidos na Seção 5. Por fim, na Seção 6, são apresentadas as conclusões e apontados alguns trabalhos futuros.

2. Aplicativos para Gerência de Contatos

As lojas online Google Play, Apple App e Windows Phone Store disponibilizam uma série de aplicativos para gerência de contatos, entretanto a grande maioria tem como objetivo facilitar a inserção, edição, organização e compartilhamento de informações sobre contatos de forma mais intuitiva para o usuário do que usando os aplicativos instalados por padrão nos sistemas operacionais Android, iOS e Windows Phone. Poucos aplicativos focam no problema da identificação e eliminação de contatos duplicados.

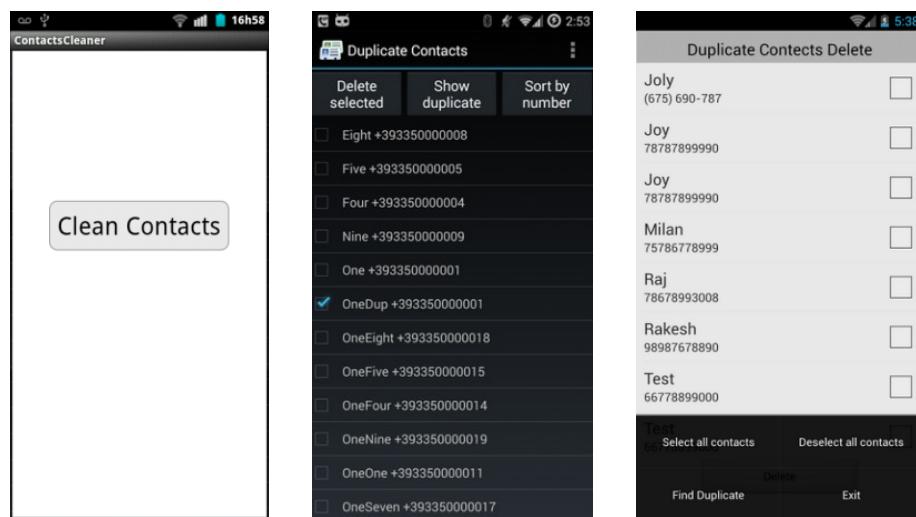


Figura 2. Interface gráfica dos aplicativos estudados que apenas eliminam contatos duplicados.

O aplicativo Limpador de Contatos [Silva 2012] remove contatos duplicados da agenda comparando apenas os números de telefone. A interface gráfica, apresentada na Figura 2 (à esquerda), possui um único botão que quando acionado remove as duplicatas sem qualquer interação do usuário. É exibida uma notificação com o número de contatos excluídos. Não é possível visualizar os contatos detectados como réplicas e tampouco restaurar a agenda original.

Já Duplicate Contacts [Accaci 2015] permite visualizar os contatos duplicados e selecionar os registros a serem excluídos. Uma seleção prévia é apresentada automaticamente para o usuário usando a igualdade dos números de telefone (vide Figura 2 ao centro). Também é possível configurar um arquivo de backup com a agenda no estado anterior às modificações.

O terceiro aplicativo estudado, denominado Duplicate Contacts Delete [Dabhi 2015], tem as mesmas funcionalidades dos apresentados anteriormente, mas utiliza, além dos números de telefone, o nome do contato para identificar duplicatas. A Figura 2 (à direita) apresenta a interface gráfica com destaque para os botões na parte inferior.

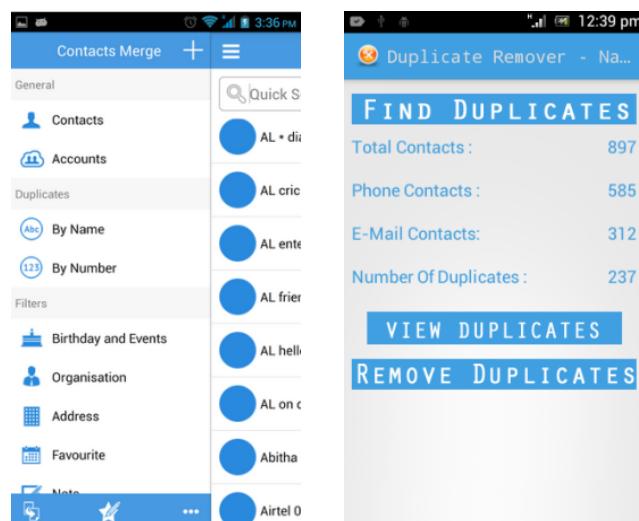


Figura 3. Interface gráfica dos aplicativos estudados que integram informações e associam contatos duplicados.

Os três aplicativos apresentados, além de serem implementados exclusivamente para o Android, permitem apenas eliminar contatos duplicados. Também foram analisados outros dois aplicativos que oferecem funções de integração das informações redundantes e associação dos contatos.

Contact Merger [ORGware Technologies 2015], disponível para Android e Windows Phone, combina todos os números de telefone de contatos com o mesmo nome, mas mantém apenas um dos nomes para contatos com o mesmo telefone. Esta segunda estratégia é perigosa porque informação relevante pode ser perdida no processo de integração. Por exemplo, o nome de um contato pode ser substituído por um apelido, como no caso de integrar “Dado” e “Eduardo Borges”. Também é possível que um qualificador de local de trabalho importante na descrição do contato seja removido, como na associação de “Renata UFRGS” e “Renata”. A interface deste aplicativo é bastante atraente (Figura 3 à esquerda) e permite o acesso a outras funcionalidades na gerência de contatos, tais como aniversários, endereços e contatos favoritos.

Por fim, Duplicate Contacts Manager [Sunil 2014] se destaca porque também utiliza o e-mail na deduplicação. Após detectar os registros duplicados, são exibidas estatísticas sobre cada tipo de contato e o número de duplicatas encontradas, que podem ser visualizadas ou removidas diretamente. A Figura 3 (à direita) apresenta a interface gráfica destacando a funcionalidade mencionada. Infelizmente, a integração está disponível apenas na versão paga e não pode ser testada. Este aplicativo está disponível apenas para o Android.

Segundo Lenzerini (2002), uma solução completa de integração de dados deve estabelecer métodos específicos que solucionem as seguintes tarefas: importar os registros de dados de diferentes fontes heterogêneas; transformar os dados de forma a obterem uma representação comum, ou seja, um esquema compatível; identificar aqueles registros semanticamente equivalentes, representando o mesmo objeto; mesclar as informações provenientes das múltiplas fontes; apresentar ao usuário final o conjunto de registros sem informação duplicada. Os aplicativos estudados concentram-se apenas nas três últimas tarefas porque utilizam métodos disponíveis na API do sistema operacional para ler os registros em um mesmo esquema.

A Tabela 1 resume as propriedades dos aplicativos estudados e as compara com as características do método proposto neste artigo. Para cada aplicativo são apresentados os *campos* utilizados na deduplicação, a função de *comparação* desses campos, o tipo de *alteração* (exclusão ou integração de contatos duplicados) e a possibilidade de restauração da agenda original.

O diferencial deste trabalho é o uso de funções de similaridade textual para comparação dos nomes e e-mails, permitindo que muitos dos casos apresentados no exemplo motivacional da Figura 1 sejam identificados como o mesmo contato. Como o foco do trabalho é o método de identificação de registros de contatos duplicados, a integração e o backup não são aplicáveis. Uma solução para associação dos contatos é um dos trabalhos futuros citados na Seção 6.

Tabela 1. Características dos aplicativos estudados e do trabalho proposto.

| Aplicativo | Campos | Comparação | Alteração | Backup |
|----------------------------|------------------------|---------------------|--------------------------|---------------|
| Limpador de Contatos | telefone | igualdade | exclusão | não |
| Duplicate Contacts | telefone | igualdade | exclusão | sim |
| Duplicate Contacts Delete | telefone, nome | igualdade | exclusão | não |
| Contact Merger | telefone, nome | igualdade | integração automática | não |
| Duplicate Contacts Manager | telefone, nome, e-mail | igualdade | interação na versão paga | sim |
| Trabalho proposto | telefone, nome, e-mail | similaridade | não aplicável | não aplicável |

3. Fundamentação Teórica

A tarefa de identificar registros duplicados que se referem a mesma entidade do mundo real é denominada deduplicação [Borges et al. 2011 b]. Nos últimos anos, diversos métodos foram propostos para a deduplicação de registros, principalmente no contexto da integração de dados relacionais [Bianco et al. 2015; Dorneles et al. 2009; Carvalho et al. 2008]. Não foram encontradas na literatura abordagens específicas para deduplicar registros de contatos em dispositivos móveis.

Grande parte dos métodos propostos para identificação de duplicatas utiliza o conceito de medida de similaridade textual, calculada através de uma função de similaridade ou de distância. As subseções seguintes apresentam e exemplificam as funções utilizadas no método proposto neste artigo [Cohen et al. 2003].

3.1. Jaccard

Sejam A e B cadeias de caracteres (*strings*) representadas por conjuntos de palavras (*tokens*). A função Jaccard calcula a similaridade entre A e B de acordo com a equação abaixo, ou seja, retorna a razão entre a quantidade de palavras compartilhadas pelas *strings* e todas as palavras que as compõem. Por exemplo, Jacard (Júlio Cesar Rodrigues, Ana Rodrigues) = $1/4 = 0,25$.

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

3.2. Levenshtein

Sejam a e b cadeias de caracteres, a distância de Levenshtein resulta no menor número de inserções, exclusões ou substituições de caracteres necessárias para transformar s em t . A similaridade é calculada como o complemento da distância normalizada, conforme

$$LevSim(a, b) = 1 - \frac{LevDist(a, b)}{\max(a, b)},$$

onde $\max(a, b)$ é o número de caracteres da maior *string*. Por exemplo, LevSim (Danilo, Daniel Rosa) = $1 - 7/11 = 0,36$.

3.3. Jaro Winkler

Seja m o número de correlações entre os caracteres e t o número de transposições, a função Jaro calcula a similaridade entre as *strings* de acordo com a equação abaixo.

$$Jaro(a, b) = \frac{1}{3} \left(\frac{m}{|a|} + \frac{m}{|b|} + \frac{m-t}{m} \right)$$

Jaro-Winkler é uma variação de Jaro que pondera prefixos, de tamanho p , em comum nas duas *strings*. Esta função é definida pela equação abaixo. Por exemplo, JaroWinkler (Eduardo Borges, Eduardo Oliveira) = $0,79 + 8/10 (1 - 0,79) = 0,93$.

$$JaroWinkler(a, b) = Jaro(a, b) + \frac{p}{10} (1 - Jaro(a, b))$$

3.4. Monge Elkan

Seja $A = \{a_1, \dots, a_K\}$ e $B = \{b_1, \dots, b_L\}$ cadeias de caracteres representadas por conjuntos de K e L palavras respectivamente. A função Monge Elkan executa para cada par de palavras uma função de similaridade auxiliar (geralmente Levenshtein) e retorna a média das máximas similaridades conforme a equação abaixo. Por exemplo, Monge Elkan (Eduardo Borges, Eduardo Oliveira) = $1/2 [\max(1, 0) + \max(0, 0,125)] = 0,56$.

$$MongeElkan(A, B) = \frac{1}{K} \sum_{i=1}^K \max_{j=1}^L sim(a_i, b_j)$$

4. Método de deduplicação proposto

A deduplicação pode ser uma tarefa bastante difícil, devido principalmente aos problemas: uso de acrônimos, diferentes estilos de formatação, estrutura dos metadados distinta, variação na representação do conteúdo, omissão de determinados campos e omissão de conteúdo relevante. Na deduplicação de contatos em dispositivos móveis não é comum o uso de acrônimos e os dados não possuem um determinado estilo. A estrutura dos registros é a mesma, porque as API dos sistemas operacionais permitem recuperar todos os registros no mesmo formato, mesmo que tenham sido coletados automaticamente de diferentes redes sociais ou outras aplicações.

Portanto, o foco da deduplicação de contatos é resolver o problema da variação e omissão de conteúdo, que é muito frequente e ainda mais grave do que em outros contextos como em bibliotecas digitais. Enquanto muitos contatos duplicados compartilham apenas o primeiro nome, referências bibliográficas apresentam diferentes representações dos nomes dos autores (ordem dos nomes e abreviações) e pouca variação no título das publicações. Além disso, contam com muitos outros metadados relevantes, como o ano e o veículo de publicação. Já a grande maioria dos contatos contam apenas com uma informação adicional além do nome: número(s) de telefone e identificador único na aplicação da qual foi coletado.

O método proposto é dividido em 3 principais fases: coleta e pré-processamento, cálculo das similaridades, e agrupamento de pares similares (vide Figura 4).

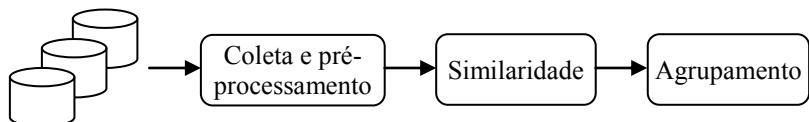


Figura 4. Método de deduplicação proposto.

Na primeira fase são coletados os contatos do dispositivo móvel provenientes da memória interna, cartão SIM e de contas vinculadas a outros aplicativos como mensageiros instantâneos e redes sociais. Para cada contato importado, são armazenados registros que contém campos que representem nome, telefone ou e-mail. Os nomes são pré-processados removendo-se acentuação, caixa alta e caracteres diferentes de letras ou números. É armazenado em um novo campo o *login* do e-mail (sem o domínio). Por fim, são mantidos apenas os 10 algarismos finais do número de telefone.

Na segunda fase os registros são combinados em pares. Aqueles que compartilham pelo menos um número de telefone ou endereço de e-mail (casamento por igualdade) são identificados como duplicados. Sobre os demais registros são aplicadas as seguintes funções de similaridade sobre seus campos: Levenshtein (logins), Jaccard (nomes), Jaro-Winkler (nomes) e Monge-Elkan (nomes). A Tabela 2 exemplifica um par de contatos e os escores retornados pelas funções.

Tabela 2. Par de contatos e os escores das funções de similaridade.

| Nome | Login | Lev | Jac | J-W | M-E |
|-----------------------|---------------|------|------|-----|------|
| Mateus Gabriel Muller | mateusmuller | 0,92 | 0,66 | 0,6 | 0,75 |
| Mateus Muller | mateusmuller2 | | | | |

Os escores de similaridade são combinados por uma média ponderada. Se o valor resultante é maior que um determinado limiar de similaridade, os registros são considerados equivalentes, ou seja, representam contatos duplicados. Os pesos e o limiar são definidos como parâmetros de configuração.

Na terceira e última fase, os pares de contatos identificados como duplicados podem ser agrupados utilizando duas estratégias diferentes: (i) cada registro é similar a pelo menos um registro do mesmo grupo; (ii) todos os registros de um grupo são similares entre si. Para implementar estas estratégias é definido um grafo de duplicatas em que cada vértice representa um contato e as arestas representam a duplicidade. Sobre este grafo são executados dois algoritmos [Kowalski & Maybury 2002]:

- *Single Link* – que retorna um grupo para cada componente conexa do grafo, implementando a primeira estratégia;
- *Click* – que retorna grupos representando subgrafos completos, implementando a segunda estratégia.

A Figura 5 ilustra o resultado dos algoritmos de agrupamento considerando o grafo de duplicatas à esquerda como entrada.

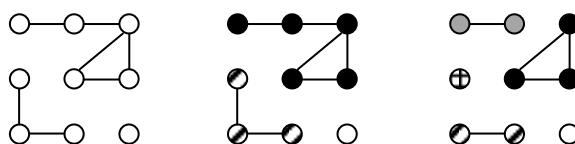


Figura 5. Exemplo de agrupamento *Single Link* (centro) e *Click* (à direita).

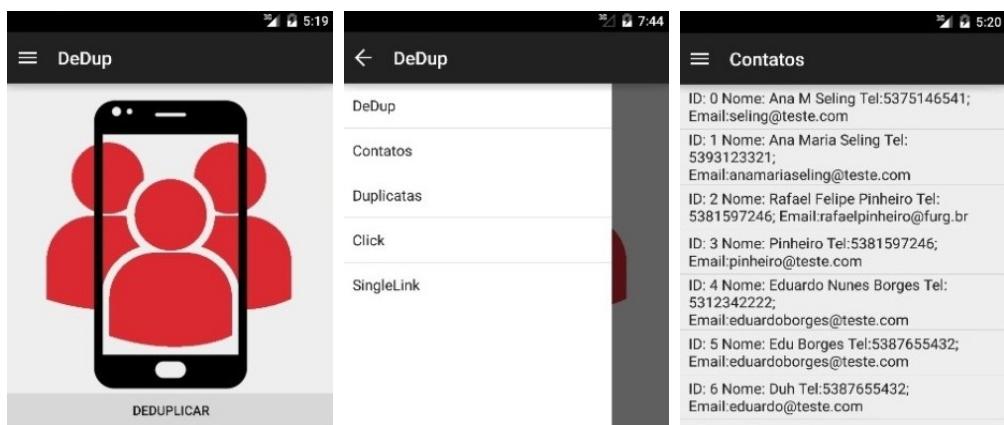


Figura 6. Interfaces do protótipo destacando a tela inicial, o menu de funções e a lista de contatos do dispositivo.

5. Avaliação Experimental

O método de deduplicação proposto foi desenvolvido na linguagem Java, utilizando o SDK do Android. A interface do protótipo é apresentada na Figura 6. São exibidas da esquerda para a direita: a tela inicial; o menu de funções e a lista de contatos do dispositivo. A Figura 7 mostra os pares identificados como duplicados junto da média ponderada dos escores retornados pelas funções de similaridade e o resultado do agrupamento dos contatos duplicados utilizando os algoritmos *Single Link* e *Click*.

Além da codificação do protótipo foi realizado um experimento para avaliar a qualidade do método proposto, através das medidas Precisão (P), Revocação (R) e F₁ [Manning et al. 2008]. Foi utilizada uma base de dados real e privada, disponibilizada por um voluntário, com exatos 1962 contatos importados de múltiplas fontes de dados: memória interna, cartão SIM, Skype, Facebook, LinkedIn, GMail e Google+.

Foram selecionadas, na primeira fase, todas as tuplas que continham pelo menos um nome, além de um número de telefone ou e-mail. Depois de finalizada a etapa de pré-processamento restaram 1072 contatos válidos.

Na segunda fase os contatos válidos foram combinados dois a dois gerando 574.056 pares. Foram excluídos pares de contatos com números de telefone ou e-mails iguais (duplicatas óbvias detectadas com casamento por igualdade), totalizando 573.044 registros, dentre os quais apenas 66 representam contatos duplicados. Para cada registro foram executadas as funções de similaridade previamente apresentadas e calculada a média variando os pesos e o limiar de similaridade da seguinte forma: $0 \leq w_{Lev} \leq 0,2$, incremento 0,1; $0,1 \leq w_{Jac} \leq 0,25$, incremento 0,05; $0,8 \leq w_{Jac} \leq 0,95$, incremento 0,05; $0,7 \leq w_{M-E} \leq 0,95$, incremento 0,05; $0,7 \leq limiar \leq 0,9$, incremento 0,1. A abrangência

| Duplicatas | SingleLink | Click |
|--|------------|-------|
| 0 1 Média: 0.62 (Calculo Similaridade) | 0 1 | 0 1 |
| 2 3 Média: 1.0 (Telefones iguais) | 2 3 | 2 3 |
| 4 5 Média: 1.0 (Email's iguais) | 4 5 6 | 4 5 |
| 5 6 Média: 1.0 (Telefones iguais) | | 6 |

Figura 7. Interfaces do protótipo destacando o resultado da deduplicação.

dos valores dos pesos foi escolhida com base na distribuição dos escores retornados pelas funções de similaridade, considerando os pares duplicados, e em resultados de experimentos anteriores com uma base de dados sintética.

A Tabela 3 resume os melhores resultados do experimento. São apresentados os pesos de cada função de similaridade no cálculo da média ponderada, o limiar de similaridade adotado, a quantidade de pares retornados (identificados como duplicados), a quantidade de pares duplicados corretamente identificados e o resultado das medidas de avaliação. Utilizando os parâmetros apresentados, identificou-se corretamente 50 dos 66 pares duplicados (Revocação máxima de 75,8%). O menor erro na deduplicação aconteceu quando foram identificados corretamente 48 dos 63 pares retornados (Precisão máxima de 76,2%), que colaborou para a melhor qualidade geral representada pela F_1 de 74,4%. Os melhores resultados não utilizaram o e-mail por isso $w_{Lev} = 0$.

Tabela 3. Resultado da avaliação experimental.

| w_{Lev} | w_{Jac} | w_{Jac} | w_{M-E} | limiar | Pares | Duplicados | P | R | F_1 |
|-----------|-----------|-----------|-----------|--------|-------|------------|--------------|--------------|--------------|
| 0 | 0,15 | 0,9 | 0,75 | 0,9 | 101 | 50 | 49,5% | 75,8% | 59,9% |
| 0 | 0,25 | 0,85 | 0,75 | 0,9 | 63 | 48 | 76,2% | 72,7% | 74,4% |

6. Considerações finais

Este trabalho apresentou um método para deduplicação de contatos que facilita o processo de integração e reduz consideravelmente o tempo em que um usuário levaria para associar manualmente contatos de diversas contas. Os experimentos realizados mostram que, utilizando funções de similaridade textual, foi possível identificar corretamente até 75,8% dos pares de contatos duplicados que não compartilham números de telefones ou endereços de e-mail. Como estes pares não podem ser detectados por nenhuma das ferramentas apresentadas na Seção 2, fica evidente a contribuição do trabalho proposto quando comparado a estas ferramentas.

Entretanto, ainda podem ocorrer erros de identificação. Por exemplo, o contato com nome = ‘Mãe’ armazenado no cartão SIM com o telefone residencial não seria detectado como duplicata do registro contendo o respectivo nome próprio e o número do celular. Ainda podem existir homônimos que não representam a mesma pessoa, como o caso de Orlando Marasciulo (registros 6, 7 e 8 da Figura 1).

Como trabalhos futuros destacam-se a avaliação da qualidade dos algoritmos de agrupamento e a criação de um algoritmo mais complexo de detecção de duplicatas que utilize técnicas de aprendizagem de máquina. Estas técnicas devem aprender com os erros e acertos dos processos de deduplicação de cada usuário de forma a aperfeiçoar o processo para os demais, configurando automaticamente os pesos e limiar de similaridade adotados como padrão. O protótipo ainda poderá ser reimplementado como um serviço local ou na nuvem para que a cada inserção ou exclusão de um contato, a deduplicação seja feita de forma incremental e bastante eficiente. A interface gráfica servirá apenas para configuração de parâmetros e interação com o algoritmo de integração, onde o usuário poderá escolher entre duas ou mais representações do nome de um contato duplicado.

Agradecimentos

Parcialmente financiado pelos Programas Institucionais de Iniciação Científica, Tecnológica e de Inovação PROBIC/FAPERGS, PIBIC-PIBITI/CNPq e PDE/FURG.

Referências

- Ableson, W. F. (2012). *Android em ação*. Rio de Janeiro: Elsevier.
- Accaci, Alex (2015). Duplicate Contacts. Disponível em <http://play.google.com/store/apps/details?id=com.accaci>. Acesso: julho de 2015.
- Bianco, G., Galante, R., Goncalves, M. A., Canuto, S., Heuser, C. A. (2015). A Practical and Effective Sampling Selection Strategy for Large Scale Deduplication. *IEEE Transactions on Knowledge and Data Engineering*, v. 27, n. 9, p. 2305-2319.
- Borges, E. N., Becker, K., Heuser, C. A. & Galante, R. (2011). A classification-based approach for bibliographic metadata deduplication. In: Proceedings of the IADIS International Conference WWW/Internet, p. 221-228, Rio de Janeiro.
- Borges, E. N., de Carvalho, M. G., Galante, R., Gonçalves, M. A., Laender, A. H. F. (2011). An unsupervised heuristic-based approach for bibliographic metadata deduplication. *Information Processing and Management*, v. 47, n. 5, p. 706-718.
- Carvalho, M. G., Laender, A. H. F., Gonçalves, M. A., da Silva, A. S. (2008). Replica identification using genetic programming. In Proceedings of the ACM Symposium on Applied Computing, p. 1801-1806. Fortaleza.
- Cohen, W., Ravikumar, P., Fienberg, S. (2003). A comparison of string metrics for matching names and records. In: KDD Workshop on Data Cleaning and Object Consolidation, v. 3, p. 73-78.
- Dabhi, Pradip (2015). Duplicate Contacts Delete. Disponível em <http://play.google.com/store/apps/details?id=com.don.contactdelete>. Acesso: julho de 2015.
- Dorneles, C. F., Nunes, M. F., Heuser, C. A., Moreira, V. P., da Silva, A. S., de Moura, E. S. (2009). A strategy for allowing meaningful and comparable scores in approximate matching. *Informaion Systems*, v. 34, n. 8, p. 673-689.
- Kowalski, G. J., Maybury, M. T. *Information Storage and Retrieval Systems : Theory and Implementation*. Springer, Boston, MA, USA, 2002.
- Lecheta, R. R. (2014). Desenvolvendo Para iPhone e iPad. Novatec.
- Lenzerini, M. (2002). Data integration: a theoretical perspective. In: Proceedings of the ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, p. 233-246.
- Manning, C. D., Raghavan, P., Schütze, H. (2008). *Introduction to information retrieval*. Cambridge: Cambridge University Press.
- ORGware Technologies (2015). Contact Merger. Disponível em <http://play.google.com/store/apps/details?id=com.orgware.contactsmerge>. Acesso: julho de 2015.
- Pinheiro, R., Lindenau, G., Zimmermann, A., Borges, E. N. (2014). Um aplicativo para integração de contatos em dispositivos Android. In: Anais do Congresso Regional de Iniciação Científica e Tecnológica em Engenharia, p. 1-4. Alegrete.
- Silva, Alan Martins (2012). Limpador de Contatos. Disponível em <http://play.google.com/store/apps/details?id=br.com.contacts.cleaner.by.alan>. Acesso: julho de 2015.
- Sunil, D M (2014). Duplicate Contacts Manager. Disponível em <http://play.google.com/store/apps/details?id=com.makelifesimple.duplicatedetector>. Acesso: julho de 2015.

aper:152853_1

Implementação de Operadores OLAP Utilizando o Modelo de Programação Map Reduce no MongoDB

Roberto Walter¹, Denio Duarte¹

¹Universidade Federal da Fronteira Sul - UFFS
Campus Chapecó

roberto.wtr@gmail.com, duarte@uffs.edu.br

Abstract. This paper presents a tool that implements OLAP operations under NoSQL DB MongoDB using Map Reduce model. The goal of this tool is to identify the performance of OLAP operations dice, slice, drill-down and roll-up executing in single and multiple nodes. Our results show that, when executing in multiple nodes, the peformance is 30% faster in the worse case.

Resumo. Este artigo apresenta uma ferramenta que implementa os operadores OLAP executados no banco de dados NoSQL MongoDB utilizando o modelo de programação Map Reduce. Os operadores implementados são: dice, slice, drill-down e roll-up. O objetivo é identificar o desempenho de tais operadores em banco de dados NoSQL executando em um nó simples e em múltiplos nós. Os resultados indicam que há um ganho de mais 30% no desempenho para o pior caso em múltiplos nós .

1. Introdução

Atualmente, há um grande e crescente volume de dados digitais [Brown et al. 2011] oriundos de diversas fontes e em vários formatos, armazenados em servidores que estão espalhados e conectados à *internet*. Bancos de dados relacionais (BDR) não foram projetados para gerenciar tais dados, pois necessitam que os dados tenham uma estrutura rígida e com pouca mudança, além da sua limitação de escalabilidade vertical. Bancos de dados NoSQL na nuvem, por outro lado, possuem arquitetura e escalabilidade horizontal para armazenar e gerenciar tais dados [Pokorny 2013]. Para a geração de informação, existem diversas ferramentas e técnicas difundidas e consolidadas para o modelo de BDR. O conjunto de operadores OLAP (*On-Line Analytical Processing*) é uma das ferramentas que está consolidada para a análise de dados estruturados. Através deles é possível realizar operações de agregação, por exemplo, e visualizar os dados de uma forma multidimensional [Chaudhuri and Dayal 1997]. No entanto, análises sobre dados semi ou não-estruturados são emergentes, e podem ser melhoradas. Sobre o processamento dos dados, uma das técnicas desenvolvidas para grandes volumes de dados é *Map Reduce*, o qual utiliza a técnica de divisão e conquista e executa de forma paralela em um *cluster* [Dean and Ghemawat 2004].

Este artigo apresenta a ferramenta *MR OLAP*, desenvolvida para a geração de operadores OLAP utilizando o modelo de programação *Map Reduce*. *MR OLAP* implementa operadores que resumem os dados em fatias (e.g. operadores OLAP *Slice* e *Dice*), e operadores que resumem e detalham as informações já obtidas (e.g. operadores OLAP *Roll-Up* e *Drill-Down*). Esses últimos operadores são implementados utilizando uma coleção que

faz o papel de dicionário de dados com informações sobre chaves que devem ser detalhadas ou sumarizadas.

Este artigo está organizado da seguinte forma: na próxima seção é apresentado o referencial teórico. A Seção 3 apresenta brevemente algumas abordagens de implementação de operadores OLAP utilizando *Map Reduce*. Na seção 4 são apresentados a estrutura de *MR OLAP*. Na Seção 5 apresenta os experimentos e a Seção 6 apresenta as conclusões.

2. Referencial Teórico

Esta seção apresenta, brevemente, os operadores OLAP, MongoDB e o modelo de programação *Map Reduce* que são as ferramentas utilizadas neste trabalho.

2.1. On-line Analytical Processing

Operadores OLAP são operadores de consultas para *data warehouses* para realizar análises sobre grandes volumes de dados [Inmon 2005]. O objetivo é facilitar a navegação sobre a estrutura do *data warehouse* e apresentar resultados dessas pesquisas de uma forma adequada. Os resultados das consultas geralmente são visualizados em formato multidimensional, compostas por dimensões que resumem uma medida. Estas estruturas são conhecidas como cubo de dados.

No modelo multidimensional, coleções de medidas referem-se aos resultados numéricos de agregação a partir do cruzamento de determinados dados. Esses dados são a relação de dimensões, que são atributos os quais as medidas são dependentes. Como exemplo, considere um caso de vendas, tal que a estrutura é definida pelas classes *Produto*, *Tempo* e *Local* como dimensões e a medida é o resultado da agregação das quantidades de venda de determinados produtos em diferentes locais e tempos. A Figura 1 apresenta esse exemplo cujas dimensões possuem os seguintes valores: *Produto* = {*Chuteira*, *Calção*, *Camisa*}, *Localidade* = {BRA, ARG, CHL}, *Tempo* = {Jan, Fev, Mar}.

| | | | | | | |
|--|-----|--|----------|------|------|------|
| | | | Camisa | | | |
| | | | Calção | | | |
| | | | Chuteira | | | |
| | BRA | | | 5300 | 6550 | 6070 |
| | ARG | | | 4010 | 4300 | 3890 |
| | CHL | | | 2340 | 3200 | 3210 |
| | | | | Jan | Fev | Mar |

Figura 1. Representações de dados no formato multidimensional.

As medidas são os valores numéricos em cada célula do cubo da Figura 1, referenciadas a partir dos eixos *x*(*Produto*), *y*(*Tempo*) e *z*(*Localidade*). Os operadores OLAP implementados neste trabalho são:

- *Drill-Down*: detalha a informação disponível, descendo a hierarquia e consultando novas dimensões.
- *Roll-Up*: operador com função inversa ao *drill-down*. Nesta operação, a informação

detalhada é summarizada.

- *Slice*: fatia o cubo, ou seja, a informação continua sendo visualizada da mesma perspectiva, no entanto, é realizado uma seleção sobre alguns dos valores do cubo.
- *Dice*: extraí um sub-cubo do cubo original. É obtido a partir de uma seleção sobre no mínimo duas dimensões.

Operadores OLAP realizam operações sobre um modelo de dados definido, cruzando diferentes dimensões e resultando medidas. Para os resultados de operadores OLAP serem satisfatórios, é importante haver uma grande amostragem de dados, para que análises de tendência e busca de padrões possam ser melhor sustentadas. Neste trabalho, os operadores OLAP são executados em um *data warehouse* modelado no banco de dados MongoDB, assunto da próxima seção.

2.2. MongoDB

MongoDB[MongoDB 2015] é um banco de dados da categoria dos NoSQL. Os bancos NoSQL foram desenvolvidos para gerenciar dados com esquema flexível, volumosos e heterogêneos, além de garantir a escalabilidade.

O banco de dados do MongoDB implementa o modelo de dados orientado a documento, ou seja, os dados são organizados na forma de documentos que representam conjuntos de chave-valor. Um conjunto de documentos forma uma coleção. Essa representação pode ser entendida no modelo relacional como: um documento representa uma tupla e uma coleção uma tabela. Para apresentar um documento é utilizado o formato JSON [Crockford 2006]. O MongoDB pode gerenciar um grande volume de dados por possuir escalabilidade horizontal, *i.e.*, composição e ligação de vários computadores (*shards*) trabalhando em conjunto (*cluster*). Juntos, os *shards* do *cluster* mantém todo o conjunto de dados do *cluster*. Tal característica, permite que operadores OLAP implementados no MongoDB, além de ter como entrada coleções volumosas de dados, possam processar os dados de forma paralela utilizando o modelo *Map Reduce*.

2.3. Map Reduce

Map Reduce é um modelo de programação criado originalmente pela Google, adequado para processar um grande volume de dados em paralelo [Dean and Ghemawat 2004], dividindo o trabalho em um conjunto de tarefas independentes.

No modelo de programação *Map Reduce*, o programador possui o trabalho de escrever duas funções, uma para o mapeamento dos dados (*map*) e outra para a redução e agregação dos dados (*reduce*). O fluxo de trabalho inclui mais duas etapas: *split* e *shuffle*. Na etapa de *split*, a entrada de dados é dividida em diversos segmentos e cada um dos segmentos é enviado para um servidor diferente do *cluster*. Baseado na definição da função, a etapa de *map* transforma os segmentos de entrada em pares <chave,valor>. Sua saída é enviada para um servidor que terá diversos segmentos de chave comum após a saída da etapa de *map*, trabalho esse realizado pela etapa de *shuffle*. Com todos os segmentos de informação comum agrupados em servidores, a função de *reduce* executa sobre esses segmentos e agrupa os pares <chave,valor> baseado na definição do usuário. A saída do *reduce* é o resultado da operação [Dean and Ghemawat 2004].

2.4. Map Reduce no MongoDB

O MongoDB provê diferentes formas para agregações. Dentre elas, estão as agregações por *pipeline*, operações de agregação de propósito único e *Map Reduce*.

O MongoDB utiliza *JavaScript* para a escrita das funções de *Map* e de *Reduce*. O *Map Reduce* no *MongoDB* é invocado pelo comando `db.<coleção>.mapreduce(<parâmetros>)`, ou `db.runCommand({mapReduce: <coleção>, <parâmetros>})`. No desenvolvimento deste artigo, foi utilizado o comando `db.runCommand`, cuja assinatura é:

```
db.runCommand( {mapReduce:<coleção>, map: <função>,
                reduce: <função>, finalize: <função>,
                out: <saída>, query: <documento>,
                sort: <documento>, limit: <numérico>,
                scope: <documento>, jsMode: <booleano>, verbose: <booleano>} )
```

Com exceção dos parâmetros *mapReduce*, *map*, *reduce* e *out*, todos os demais são opcionais. A seguir, são especificados os parâmetros que foram utilizados no desenvolvimento da ferramenta *MR OLAP*:

- *mapReduce*: nome da coleção de entrada para o *Map Reduce*;
- *map*: função *JavaScript* para fazer o mapeamento dos pares `<chave,valor>`;
- *reduce*: função *JavaScript* que reduz (agrega) os valores do par `<chave,valor>`;
- *query*: critério de delimitação dos documentos de entrada da função *map*; e
- *out*: definição de saída do *Map Reduce*. A saída pode ser uma coleção com os documentos gerados ou na saída padrão conforme definido em *out* (geralmente a tela, *stdout*);

3. Trabalhos Relacionados

A partir do uso de *Map Reduce*, algumas propostas de desenvolvimento para atender às necessidades de operadores *OLAP* foram desenvolvidas. Nesta seção são descritas as abordagens *Full Source Scan*, *Index Random Access*, *Index Filtered Scan* [Abelló et al. 2011] e *MRPipelevel* [Lee et al. 2012].

A primeira abordagem consiste no algoritmo *Full Source Scan* (FSS), algoritmo baseado na força bruta para leitura dos dados, implementado através de paralelismo. Primeiramente, configura-se a leitura dos dados para que somente as colunas de interesse do cubo final (*i.e.*, dimensões, medidas) sejam retornadas. Após, são excluídos os pares `<chave,valor>` que não atendem ao filtro proposto. A função de *Map* redefine as chaves para a dimensão do cubo de saída, e o valor como a medida que será agregada. Depois de todos os pares `<chave,valor>` gerados pelo script de *Map* terem sido agrupados pela chave, a função de *reduce* é invocada uma vez para cada valor de dimensão. Assim, o *reduce* apenas precisa replicar a chave da entrada na saída e agregar os valores da medida correspondente àquele registro.

A abordagem de *Index Random Access* (IRA) é uma melhoria do algoritmo *Full Source Scan* e parte do pressuposto de que varrer os arquivos deve ser evitado (*Full Scan*). Assim, busca-se utilizar alguma técnica de indexação para evitar um *full scan* toda vez em que necessita-se gerar algum relatório. Assim, após a leitura dos arquivos, é construído uma estrutura de índice. Há mais duas fases que são executadas quando um cubo de

fato necessitar ser construído. Na primeira fase, os *slicers* são carregados e a estrutura de índices é acessada de forma aleatória. A última fase faz um *scan* da saída e acessa a tabela de dados para todo ID que foi encontrado na estrutura de índices [Abelló et al. 2011]. No *reduce*, os valores dos dados finais são agregados.

O algoritmo *Index Filtered Scan* é uma adaptação da abordagem *Index Random Access*. O intuito dessa proposta é utilizar a estrutura de índice do método IRA, mas evitar o acesso aleatório à fonte de dados. Após a criação da estrutura de índices no algoritmo IRA, é criado um *bitmap* na memória baseado nos IDs da estrutura de índices. As colunas de dimensões e medidas são elencadas para a agregação e uma leitura sobre os dados temporários é executada. Após a leitura dos dados temporários e a geração do *bitmap*, o mesmo é utilizado para filtrar os pares gerados. Finalmente, o *reduce* realiza a agregação das medidas. Assim, evita-se o acesso aleatório aos dados indexados, tornando o algoritmo mais eficiente na etapa de busca dos dados na estrutura de índice.

Por último, a proposta de *MRPipeLevel* é um algoritmo baseado no algoritmo *PipeSort* [Agarwal et al. 1996], o qual gera árvores ordenadas de custo mínimo a partir de valores quantitativos das dimensões. No *PipeSort*, vários níveis de informação são computados ao mesmo tempo, sem ter a necessidade de que informações entre níveis dependentes estejam no mesmo processo de agregação. Isso é tratado pela ordenação dos dados antes da agregação, assim evita-se realizar leitura de dados de forma repetida. A Figura 2 apresenta um exemplo de agregação pelo *pipeline*. Considere que as colunas A, B e C são dimensões, e o objetivo é contar quantas combinações de valores há dessas três dimensões. Nesta situação, a contagem de combinações sobre as dimensões AB, A e all, que refere-se à todas as dimensões, pode ser realizada durante a contagem sobre a combinação das dimensões ABC. Considere que o resultado é estruturado por $\langle A \ B \ C, \text{contagem} \rangle$. A contagem das combinações na primeira tupla resulta em ABC: $\langle 1 \ 1 \ 1, 1 \rangle$, AB: $\langle 1 \ 1 \ *, 1 \rangle$, A: $\langle 1 \ * \ *, 1 \rangle$, all: $\langle * \ * \ *, 1 \rangle$. Na segunda tupla resulta ABC: $\langle 1 \ 1 \ 1, 2 \rangle$, AB: $\langle 1 \ 1 \ *, 2 \rangle$, A: $\langle 1 \ * \ *, 2 \rangle$, all: $\langle * \ * \ *, 2 \rangle$. Na agregação da terceira tupla, $\langle 1 \ 1 \ 1, 2 \rangle$ é emitido como um resultado de ABC, pois como os dados estão ordenados, a combinação $\langle 1 \ 1 \ 1 \rangle$ de ABC não ocorrerá novamente. As demais agregações na terceira tupla são ABC: $\langle 1 \ 1 \ 3, 1 \rangle$, AB: $\langle 1 \ 1 \ *, 3 \rangle$, A: $\langle 1 \ * \ *, 3 \rangle$ e all: $\langle * \ * \ *, 3 \rangle$. Na quarta tupla, $\langle 1 \ 1 \ 3, 1 \rangle$ e $\langle 1 \ 1 \ *, 3 \rangle$ são emitidos como resultado de ABC e AB respectivamente. Os demais resultados da contagem na quarta tupla são AB: $\langle 1 \ 2 \ *, 1 \rangle$, A: $\langle 1 \ * \ *, 4 \rangle$ e all: $\langle * \ * \ *, 4 \rangle$. Utilizando uma agregação *pipeline* dessa forma, os resultados de ABC, AB, A e all podem ser computados juntos e realizam a leitura da entrada de dados apenas uma vez.

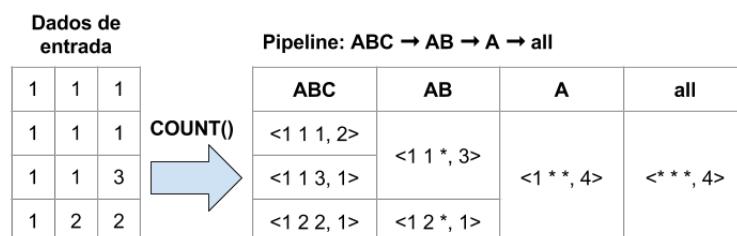


Figura 2. Exemplo de agregação pipeline [Lee et al. 2012]

4. Proposta

A ferramenta *MR OLAP* é baseada na proposta do Algoritmo *Full Source Scan* (*FSS*) [Abelló et al. 2011]. A opção pelo *Full Source Scan* ocorre pelo fato deste ser a base para desenvolvimento das otimizações *Index Random Access* e *Index Filtered Scan*. Assim, poderá ser extendida para utilizar as otimizações.

As entradas de *MR OLAP* são as seguintes:

1. Uma coleção de dados Δ no formato *JSON*;
2. Uma lista de dimensões D , aonde D é um subconjunto de chaves da coleção Δ . D compõe a chave key_{mr} que é utilizada na execução de *Map Reduce*. Primeiramente $key_{mr} = D$, no entanto, dependendo do operador a ser desenvolvido, key_{mr} sofrerá variações e poderá ser um subconjunto ou superconjunto de D .
3. Uma medida M que refere-se à chave de Δ definida para ser agregada na função de *Map Reduce*.
4. Uma lista F , tal que $F_{chaves} \subset D$, de possíveis filtros (opcional);
5. O operador OLAP O_L a ser gerado, sendo que O_L pode ser *Dice*, *Slice*, *Drill-Down* e *Roll-Up*;
6. Um atributo C (condicional) a ser detalhado, caso O_L corresponda ao operador *Drill-Down*, ou sumarizado, caso O_L corresponda ao operador *Roll-Up*;

A geração do cubo OLAP pela ferramenta *MR OLAP* ocorre em 4 etapas:

1. Leitura das chaves $D \cup M$ por meio de uma varredura dos dados em Δ .
2. Mapeamento das dimensões D e composição da chave key_{mr} .
3. Redução e agregação dos dados da medida M . O resultado é gravado na coleção Δ_{temp} ;
4. Montagem do operador OLAP conforme O_L ;

Definição 1 - Estrutura de Hierarquização: *Dadas uma coleção rw_structure, composta pelas chaves colecao e estrutura, aonde colecao refere-se à uma coleção Γ a ter sua estrutura de hierarquização de chaves definida, e estrutura refere-se às estruturas de hierarquização das chaves λ de Γ . A estrutura de hierarquização é definida pela separação das chaves λ pelo caractere ";". Portanto, uma estrutura de hierarquização é definida como $\lambda_1;...;\lambda_{n-1};\lambda_n$, aonde a chave λ mais à esquerda (λ_1) refere-se ao maior nível de detalhe da estrutura, e a chave λ mais à direita (λ_n) refere-se ao maior nível de sumarização da estrutura. Existe a possibilidade de uma coleção possuir mais de uma estrutura de hierarquização, assim, para separar as diferentes estruturas, na chave estrutura as estruturas são separadas pelo caractere especial "#".*

Suponha uma estrutura de hierarquização E obtida através da Definição 1, suponha uma chave $C \in E$. A Definição 2 formaliza a obtenção da chave *chave sumário*.

Definição 2 - $C_{sumario}$: *Uma chave sumário ($C_{sumario}$) é a chave que aparece imediatamente à direita da chave C na hierarquia E .*

Suponha uma estrutura de hierarquização E obtida através da Definição 1, suponha uma chave $C \in E$. A Definição 3 formaliza a obtenção da chave *chave detalhe*.

Definição 3 - $C_{detalhe}$: *Uma chave detalhe ($C_{detalhe}$) é a chave que aparece imediatamente à esquerda da chave C na hierarquia E .*

A seguir são descritos os possíveis direcionamentos da etapa 4 do algoritmo:

1. Caso $O_L = slice$: o filtro F é aplicado com o operador de igualdade na coleção Δ_{temp} , a fim de que seja selecionada uma fatia de dados de Δ_{temp} .
2. Caso $O_L = dice$: o filtro F , com operador lógico definido pelo usuário, é aplicado sobre a coleção Δ_{temp} . No operador *Dice* pode ser aplicado uma lista de filtros.
3. Caso $O_L = roll-up$: o comportamento dos filtros é o mesmo que em *Dice*. Em *Roll-Up* é realizada a etapa de sumarização, onde a chave C é removida da coleção Δ_{temp} e da chave key_{mr} . A chave $C_{sumario}$ (obtida a partir da Definição 2), correspondente à sumarização de C , substitui C na composição de D e key_{mr} . Com a nova estrutura de D e key_{mr} , agora com $C_{sumario}$ no lugar de C , um novo *Map Reduce* é executado sobre D e key_{mr} .
4. Caso $O_L = drill-down$: o conjunto de dados processado passa a ter um nível de detalhe a mais. O operador *Drill-Down* é semelhante ao operador *Roll-Up*, com a diferença de que a chave C não é removida da coleção Δ_{temp} para ser sumarizada, mas a mesma é mantida nas estruturas D e key_{mr} e passa a ser detalhada. O detalhamento ocorre pela geração de um ou mais registros $R_{detalhe}$ abaixo de cada registro de Δ_{temp} . Os registros $R_{detalhe}$ correspondem aos mesmos dados do registro de Δ_{temp} , mas com a adição da chave $C_{detalhe}$ (obtida a partir da Definição 3), correspondente a uma chave de detalhamento de C . Para a geração de cada registro de $R_{detalhe}$, é criado uma chave $key_{detalhe}$, tal que $key_{detalhe} = key_{mr} \cup C_{detalhe}$, e executado um *Map Reduce* com a chave $key_{detalhe}$ e valor sendo o mesmo da medida agregada para a geração dos registros de Δ_{temp} .

Após a execução das 4 etapas de *MR OLAP*, é realizado uma formatação do *layout* de Δ_{temp} e o resultado final é atribuído à coleção C_R .

Assim, como proposto no Algoritmo *Full Source Scan*, *MR OLAP* propõem a implementação de operadores que resumem em fatias os dados analisados. Porém, além dos operadores *Slice* e *Dice*, foram implementados os operadores de sumarização e detalhamento, respectivamente, *Roll-Up* e *Drill-Down*. Com estes operadores, é permitido ao usuário navegar entre os níveis dos dados que podem partir do mais sumarizado ao mais detalhado, e vice-versa. Para que seja possível verificar as hierarquias de chaves para detalhamento e sumarização, foi disponibilizada a coleção *rw_structure* para servir de metadado sobre os dados de hierarquização.

5. Experimentos

Esta seção apresenta os resultados dos experimentos realizados. Os dados do estudo de caso foram extraídos de uma base de dados Oracle[®], onde atributos de diferentes tabelas foram reunidos, convertidos para JSON e gravados na coleção *rw_fato_vendas* do MongoDB. A coleção *rw_fato_vendas* é composta pelas chaves *_id*, *pedido*, *cod_produto*, *modelo*, *data_entrega*, *ano*, *cidade*, *uf*, *pais*, *marca* e *quantidade*. A chave *quantidade* foi alterada para garantir o sigilo das informações do *Grupo Dass*, fornecedor da base de dados. Os dados referem-se as vendas do *Grupo Dass* que ocorreram desde o ano 2000 de produtos com estoque no Brasil. Para os experimentos foi preparado um *cluster* com a instalação do MongoDB 3.0 em 4 computadores com sistema operacional Ubuntu Linux 15.10 (64 bits), processador Intel Core i5 - 3470 @ 3,20 GHz com 8GB de memória RAM. Os experimentos foram realizados no *cluster* com 4 computadores, sendo que 3

processam paralelamente, e também com processamento em uma única máquina, a fim de comparar os desempenhos.

Para o experimento, considere que o usuário gostaria de ter um relatório com a quantidade total de vendas com previsão de entrega para o ano de 2014. Para isso, são considerados os seguintes cenários:

C1 (operador *Slice*): selecionar as dimensões modelo, marca, UF e ano, e aplicar o filtro sobre a dimensão ano igual a 2014.

C2 (operador *Dice*): selecionar as dimensões modelo, marca, UF e ano, e aplicar o filtro sobre a dimensão ano igual a 2014 e marca igual a TRYON.

C3 (operador *Roll-Up*): selecionar as dimensões modelo, marca, UF e ano, e aplicar o filtro sobre a dimensão ano igual a 2014 e marca igual a TRYON. Além disso, o usuário seleciona a dimensão UF para ser sumarizada, assim a quantidade total será sumarizada e consolidada por PAIS.

C4 (operador *Drill-Down*): selecionar as dimensões modelo, marca, UF e ano, e aplicar o filtro sobre a dimensão ano igual a 2014 e marca igual a TRYON. Além disso, o usuário seleciona a dimensão UF para ser detalhada, assim para cada registro de total por UF será detalhado o total de cada cidade da UF.

Dentro do cenário proposto, o *cluster* com quatro computadores/*shards* foi denominado de 1 e o *cluster* com um computador foi denominado de 2. Assim, o cenário *C1* executado na configuração 1 é denominado de *C1.1*. O mesmo raciocínio segue para os outros casos. Por exemplo, a execução no cenário 3 com a configuração de *cluster* 2, é chamada de *C3.2*. Além disso, em cada cenário/configuração foi definido o tamanho das coleções de entrada. Neste experimento, foram considerados dois tamanhos: 4.401.849 de documentos (chamado de 1) e 1.428.313 (chamado de 2). Assim, um dado experimento foi composto e nomeado com os três tipos de configuração. Por exemplo, o Cenário 2 (*C2*), com a configuração 2 e o tamanho de coleção 1, foi denominado de *C2.2.1*. A métrica resultante do cenário foi o tempo de execução necessário para finalizar a geração do operador OLAP. Cada cenário foi executado três vezes e o tempo resultante foi obtido pelo cálculo da média aritmética das três execuções. A Tabela 7.1 apresenta resultados obtidos na geração dos operadores OLAP sob os cenários apresentados. A primeira coluna identifica a configuração do cenário descrito anteriormente. A segunda coluna apresenta o tempo de processamento necessário para gerar a configuração da primeira coluna. Por fim, a terceira coluna indica o percentual de melhora de desempenho da configuração com 1 *shard* para 3 *shards*. O desvio padrão não foi informado pois não houve variação significativa entre os tempos calculados.

Com base nas informações da Tabela 1, pode-se afirmar que o tempo de geração do operador OLAP depende diretamente do número de *shards* utilizados para processamento e depende também do tamanho da coleção de entrada. Por exemplo, são necessários 2 min, 57 seg e 662 ms para processar 4.401.848 documentos em 3 *shards* na geração do operador *Slice* (*C1.1.1*), mas esse tempo sobe para 4 min, 41 seg e 101 ms quando essa configuração é aplicada em apenas 1 *shard* (*C1.2.1*), isso indica que *C1.1.1* teve um desempenho 36.79% melhor comparado a *C1.2.1*. Se comparado com número de documentos igual a 1.428.313, o tempo de execução de 2 min, 3 seg e 103 ms em *C1.1.2* aumenta para 4 min, 39 seg e 282 ms em *C1.2.2*, uma melhora de 55.92% quando processado em 3 *shards* comparado a 1. Para os demais operadores verificou-se que há

| Configuração | Tempo | Δ desempenho |
|--------------|------------------------|---------------------|
| C1.1.1 | 2 min, 57 seg, 662 ms | 36.79% |
| C1.2.1 | 4 min, 41 seg, 101 ms | |
| C1.1.2 | 2 min, 3 seg, 103 ms | 55.92% |
| C1.2.2 | 4 min, 39 seg, 282 ms | |
| C2.1.1 | 2 min, 55 seg, 57 ms | 35.57% |
| C2.2.1 | 4 min, 31 seg, 733 ms | |
| C2.1.2 | 1 min, 59 seg, 812 ms | 56.17% |
| C2.2.2 | 4 min, 33 seg, 398 ms | |
| C3.1.1 | 4 min, 30 seg, 991 ms | 33.87% |
| C3.2.1 | 6 min, 49 seg, 845 ms | |
| C3.1.2 | 2 min, 18 seg, 781 ms | 70.65% |
| C3.2.2 | 7 min, 52 seg, 923 ms | |
| C4.1.1 | 28 min, 25 seg, 463 ms | 55.44% |
| C4.2.1 | 63 min, 47 seg, 891 ms | |
| C4.1.2 | 12 min, 23 seg, 681 ms | 58.11% |
| C4.2.2 | 29 min, 35 seg, 686 ms | |

Tabela 1. Resultados dos experimentos.

um padrão de melhor desempenho diretamente proporcional ao número de *shards* processando paralelamente, e inversamente proporcional ao número de documentos a serem processados. Foi confirmado que o paralelismo significou ganho de desempenho e que o número de documentos da coleção de entrada influenciou no tempo de execução.

Os algoritmos para geração de *Slice* e *Dice* utilizam um *Map Reduce*, e para *Roll-Up* utiliza duas execuções de *Map Reduce*, assim, estes algoritmos possuem complexidade assintótica $O(n)$ para o tamanho da entrada (número de documentos da coleção). Percebe-se através da tabela que a maior parte dos experimentos possui um tempo de geração abaixo de 8 minutos, sendo que, apenas nos experimentos do operador *Drill-Down* esse tempo é excedido. O algoritmo desenvolvido para a geração do operador *Drill-Down* utiliza um primeiro *Map Reduce*, mas para cada documento resultante é gerado um novo *Map Reduce* sobre todos os dados da coleção na busca por documentos para o detalhamento do documento atual. Essa implementação possui complexidade assintótica $O(n^2)$ para o tamanho da entrada. Devido a este fato, as gerações de *Drill-Down* possuem um tempo de execução mais elevado comparado aos demais operadores. No entanto, assim como para os demais operadores, na geração do operador *Drill-Down* obteve-se ganho de desempenho quando o número de *shards* processando aumentou. Além da medição dos tempos necessários para geração dos operadores, foi feita uma verificação manual de corretude dos dados gerados, e concluiu-se que as respostas dos algoritmos estão corretas.

6. Conclusão

Ferramentas de análise devem estar preparadas para a manutenção e extração de informações de um grande conjunto de dados que podem estar em diferentes formatos. Operadores OLAP são uma solução para análise de dados para bancos de dados tradicionais. No entanto, ainda pode-se avançar em melhorias dessas ferramentas OLAP para bancos da classe NoSQL. Neste contexto, este trabalho apresentou a ferramenta *MR OLAP*, uma ferramenta que realiza a geração de operadores OLAP utilizando o modelo de programação *Map Reduce*.

Os experimentos mostraram que a execução dos operadores possui maior desem-

penho quando são processados paralelamente em mais de um *shard*. O tamanho da base de dados influencia diretamente no desempenho para a geração dos operadores, principalmente para o operador *Drill-Down* cujo algoritmo de geração possui complexidade $O(n^2)$. O tamanho da base de dados influencia os operadores *Slice*, *Dice* e *Roll-Up* somente no tempo de agrupamento e agregação dos dados. Então o desempenho da geração dos operadores diminui linearmente na medida em que a base de dados cresce. Para o operador *Drill-Down*, por outro lado, o desempenho apresenta um decrescimento quadrático na medida em que a base de dados cresce. *MR OLAP* cumpre seu propósito, gerando operadores *OLAP* de forma mais eficiente processando paralelamente em mais *shards*. Em [Walter 2015], *MR OLAP* é descrito com mais detalhes.

Assim, têm-se as seguintes perspectivas de continuação desse trabalho: (i) aprimorar a implementação dos operadores para melhorar seu desempenho, (ii) diminuir a complexidade do algoritmo de geração do operador *Drill-Down*, (iii) desenvolver algoritmos para agregar novos operadores *OLAP* na ferramenta, (iv) possibilitar a comparação de resultados entre operadores, (v) adicionar recursos visuais para a análise dos resultados em gráficos, e (vi) desenvolver os operadores com base em outra abordagem dos trabalhos relacionados, a fim de comparar o desempenho entre as implementações. Existem perspectivas de melhorias da ferramenta *MR OLAP*, mas é importante ressaltar que atualmente a ferramenta realiza a geração dos operadores *OLAP* conforme pretendido.

Referências

- Abelló, A., Ferrarons, J., and Romero, O. (2011). Building cubes with MapReduce. In *Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP*.
- Agarwal, S., Agrawal, R., Deshpande, P. M., Gupta, A., Naughton, J. F., Ramakrishnan, R., and Sarawagi, S. (1996). On the computation of multidimensional aggregates. In *VLDB*.
- Brown, B., Chui, M., and Manyika, J. (2011). Are you ready for the era of ‘big data’. *McKinsey Quarterly*, 4:24–35.
- Chaudhuri, S. and Dayal, U. (1997). An overview of data warehousing and OLAP technology. *ACM Sigmod record*, 26(1):65–74.
- Crockford, D. (2006). The application/json media type for javascript object notation (JSON).
- Dean, J. and Ghemawat, S. (2004). Mapreduce: simplified data processing on large clusters. *Communications of the ACM*.
- Inmon, W. H. (2005). *Building the data warehouse*. John wiley & sons.
- Lee, S., Kim, J., Moon, Y.-S., and Lee, W. (2012). *Efficient distributed parallel top-down computation of rolap data cube using MapReduce*. Springer.
- MongoDB (2015). The MongoDB 3.0 manual.
- Pokorny, J. (2013). NoSQL databases: a step to database scalability in web environment. *International Journal of Web Information Systems*, 9(1):69–82.
- Walter, R. (2015). Implementação de operadores OLAP utilizando o modelo de programação Map Reduce. TCC, UFFS (cc.ufffs.edu.br/tcc).

aper:152932_1

Mineração de dados para modelos NoSQL: um survey

Fhabiana Thieli dos Santos Machado¹, Deise de Brum Saccol²

¹Programa de Pós Graduação em Informática – Universidade Federal de Santa Maria (UFSM)
Santa Maria – RS – Brasil

²Departamento de Linguagens e Sistemas de Computação – UFSM
Santa Maria – RS – Brasil

fsantos@inf.ufsm.br, deise@inf.ufsm.br

Abstract. Recent years have witnessed the emergence of new models of databases. These models, known as NoSQL, are characterized by not having a formal structure, not providing access via SQL, being distributed and promising greater scalability and performance. When popularized, these models originated a gap in terms of data analysis, since the data mining tools were usually designed to be applied to relational models, not to the unstructured or semi-structured data. To that end this paper performs a search involving data mining to semi-structured/unstructured data limited to the scope of possible data formats stored in NoSQL.

Resumo. Nas últimas décadas houve o surgimento de novos modelos de bases de dados. Estes modelos, conhecidos como NoSQL, se caracterizam por não possuírem uma estrutura formal, não fornecerem acesso via SQL, serem distribuídos e prometerem maior escalabilidade e desempenho. Ao se popularizarem originaram uma lacuna em termos de análise de dados, visto que as ferramentas de mineração de dados, por exemplo, usualmente foram desenvolvidas para serem aplicadas a modelos relacionais, não a dados sem estrutura ou semi-estruturados. Nesse intuito o presente trabalho realiza uma pesquisa em mineração de dados semi-estruturados/ não-estruturados limitados ao escopo dos possíveis formatos de dados armazenados em NoSQL.

1. Introdução

Nos últimos anos surgiram diferentes modelo de bancos de dados. Dentre estes estão os caracterizados por não serem relacionais, possuírem esquema livre e executarem de forma distribuída. Tais modelos são denominados "NoSQL"(Not only Structured Query Language) [Sadlage and Fowler 2012]. Neste contexto cada solução foi desenvolvida para uma necessidade específica e não há uma padronização.

Por outro lado, o processo de descoberta de conhecimento que visa extrair informações não triviais de bases de dados não está preparado para suprir essa nova demanda. Este envolve etapas como a de limpeza dos dados, integração, seleção, transformação, mineração, avaliação e apresentação [Han et al. 2011]. Neste ponto há uma lacuna visto que, por exemplo, essas técnicas usualmente foram desenvolvidas para dados estruturados.

Sendo assim, a diversidade dos modelos NoSQL trouxe grandes desafios para a mineração de dados, como por exemplo a de trabalhar com tipos complexos de dados.

Conforme [Han et al. 2011] há um amplo espectro de novos tipos que vão desde os dados estruturados dos modelos relacionais a semi-estruturados e dados não estruturados de repositórios dinâmicos. Dessa forma a presente pesquisa tem por objetivo analisar trabalhos relevantes envolvendo mineração de dados semi-estruturados/ não estruturados limitados ao escopo dos formatos de dados armazenados em bancos de dados NoSQL.

É importante ressaltar que o objetivo não é abordar toda a literatura referente a mineração de dados semi-estruturados/ não estruturados, mas sim aqueles que se enquadram no escopo de NoSQL. Utilizando pesquisa avançada na base IEEE Xplore¹ com as palavras-chave (*NoSQL and mining and knowledge*) apenas 16 artigos são retornados. Além destes, aproximadamente outros 352 resultados apontados no Scholar Google² resultantes da pesquisa com os termos (*nosql +mining +unstructured +knowledge -cloud*) são considerados. Para fins de análise, foram selecionados cerca de 10 publicações cujo escopo estava de acordo com o critério proposto da pesquisa e também segundo o número de citações.

Uma das principais classificações adotadas para NoSQL é a proposta por [Sadlage and Fowler 2012] sobre os modelos de dados chave-valor, documentos, colunar e de grafos. Neste sentido, a abordagem proposta analisa os trabalhos sob o aspecto dos formatos que podem ser armazenados nestes bancos de dados. Assim sendo, classificados em: texto, documentos ou grafos.

O presente trabalho segue estruturado da seguinte forma: primeiramente a Fundamentação Teórica (Seção 2) com os conceitos de modelo de dados NoSQL e Descoberta de Conhecimento. Logo a explanação sob os aspectos abordados de mineração, sendo Mineração de Textos (Seção 3), Mineração em Documentos (Seção 4) e Mineração em Grafos (Seção 5). Após isto, Discussões (Seção 6) e por fim a Conclusão (Seção 7).

2. Fundamentação teórica

Um dos principais conceitos abordados neste trabalho diz respeito ao tipo de dado, o qual pode ser estruturado, semi-estruturado e não estruturado. O primeiro é aquele que possui estrutura formal. O segundo não possui estrutura rígida mas pode possuir marcas ou *tags* segundo [Kanimozhi and Venkatesan 2015], como XML ou JSON. O terceiro, contudo, não possui estrutura alguma e conforme [McKendrick 2011] pode se referir a: documentos comerciais, PDF, conteúdo de redes sociais, artigos digitalizados, vídeo, áudio, conteúdo web, dentre outros.

2.1. Modelos de dados NOSQL

Não há uma única definição para o termo NoSQL, mas conforme [Begoli 2012] é caracterizado pelas bases de dados que não oferecem semântica SQL, nem propriedades ACID (Atomicidade, consistência, isolamento e durabilidade), bem como apresentam arquitetura distribuída e tolerante à falhas. Geralmente são classificados por seu modelo de dados como [Sadlage and Fowler 2012] aponta: chave-valor, documentos, colunar e de grafos. Para efeito de estudo será abordado o aspecto relacionado ao formato de armazenamento de dados desses modelos.

¹<http://ieeexplore.ieee.org/search/advsearch.jsp>

²<https://scholar.google.com.br/>

No modelo chave-valor são armazenados registros com pares chave e valor. No geral seus atributos podem ser somente do tipo "*String*", assim como no Amazon SimpleDB [Padhy et al. 2011]. Dessa forma esses dados serão considerados no aspecto de texto do trabalho.

Por outro lado, em uma base de dados de documentos se armazenam coleções de documentos compostas por campos. Geralmente seus componentes são armazenados em formato JSON, como é o caso de CouchDB e MongoDB. Para fins de estudo, estes serão considerados como dados semi-estruturados, ou seja, em aspecto de documento aqueles que trabalham com formato JSON, XML ou outra linguagem de marcação.

O modelo colunar é semelhante a um mapa cuja estrutura é formada por linha, coluna e *timestamp* que compõem uma chave. Demonstra [Padhy et al. 2011] que é composto por uma linha (*String*), coluna (*String*), *timestamp* (*int* de tamanho 64) e o resultado em *String*, sendo considerado no aspecto de texto.

Bancos de dados em grafo em sua essência visam armazenar dados de forma a auxiliar em consultas [Lomotey and Deters 2014c], bem como, melhor estabelecer relacionamento entre eles. Como este último modelo possui uma estrutura diferente, ou seja, um formato que envolve nós, arestas e linhas, será considerado um aspecto separado, o de grafos.

2.2. Descoberta de conhecimento

De acordo com [Begoli 2012] a descoberta de conhecimento em dados (do inglês *Knowledge Discovery from Data*) é um conjunto de atividades destinadas a extrair novos conhecimentos de conjuntos de dados grandes e complexos.

O processo de KDD envolve as etapas de limpeza dos dados (responsável por eliminar ruídos e inconsistências), integração (de múltiplas fontes), seleção (dados relevantes), transformação (através de operações de resumo e agregação), mineração (processo essencial), avaliação e apresentação (visualização).

Com relação à mineração de dados as principais tarefas utilizadas são descrição, estimativa, previsão, classificação, agrupamento e regras de associação sendo que algumas são aplicadas a dados numéricos e outras a dados qualitativos, ou não-numéricos.

Outra estratégia utilizada como base para a etapa de mineração é a de *Data Warehouse* que segundo [Han et al. 2011] são repositórios orientados a assunto, tempo, integrados e auxiliam na tomada de decisões. Uma das técnicas de análise que se pode aplicar a um modelo de dados multidimensional é OLAP (*Online Analytical Processing*). Com esta técnica eles podem ser organizados em diferentes níveis de detalhe, podendo navegar para cima (*roll-up*) e para baixo (*drill-down*), além de girar (*rotate*) e cortar (*slice-and-dice*).

3. Mineração em texto

O campo da mineração de textos é bem vasto. Conforme apontado em trabalhos de [Lomotey and Deters 2014a] há um leque de técnicas propostas como: algoritmos baseados em recuperação de informação, regras de associação, tópicos e *topic maps*, termos, *cluster* de documentos, entre outras. Esta é uma área de pesquisa bem madura no que diz

respeito a aplicações tradicionais. O formato de armazenamento em texto é suportado por praticamente todos os modelos NoSQL, sendo portanto, o mais abrangente.

3.1. Abordagem baseada em ferramentas

Existem algumas ferramentas proprietárias para mineração de textos que estejam contidos em conteúdos da web, livros, comentários de *blogs*, etc. Dentre as principais ferramentas estão *Apache Mahout*, *SAS Text Miner* e demais relacionadas com a linguagem R, porém não são aplicáveis diretamente a NoSQL.

Há a possibilidade de adaptá-las. Como exemplo, [Chakraborty 2014] utiliza *SAS Text Miner* para analisar um coleção de documentos propondo uma nova forma de organizar e analisar dados textuais. Isto é feito aplicando técnicas de mineração como modelo de regressão em conjunto com redes neurais artificiais para tentar prever uma variável-alvo.

Mineração de texto também pode ser encontrada como mineração de opinião no campo de análise de sentimentos. Geralmente aplicadas em redes sociais para determinar preferências positivas, negativas ou neutras sobre produtos, como em [Kim et al. 2012]. Ao extrair palavras-chave do Twitter filtradas através de processamento de linguagem natural, utiliza OLAP em conjunto com R para inferir preferência de usuários sobre celulares.

Outras ferramentas podem ser exploradas em conjunto com técnicas específicas para se buscar extrair conhecimento de dados texto armazenados em uma base de dados NoSQL qualquer.

3.2. Abordagem baseada em framework

Outra alternativa encontrada é utilizar uma base de dados NoSQL como parte de uma arquitetura maior. O propósito é de armazenar os dados de forma a aproveitar a escalaabilidade e até a possibilidade de distribuí-los para melhor desempenho em acesso e recuperação dos mesmos. Suas aplicações podem abranger diferentes áreas.

A abordagem de [Wylie et al. 2012] propõe analisar sistemas de *streaming* de dados em tempo real com tempo de resposta com menos de um minuto. Neste caso o NoSQL CouchDB é parte central do modelo que se comunica com demais módulos. A principal utilização de seu sistema é para a realização de *streaming* de atividades em análise de texto e classificação em um link de rede que recebe cerca de 200 *e-mails* por dia.

Outrossim, pode ser aplicado a um sistema que utiliza ontologias para inferir conhecimento [Liu et al. 2015]. Em seu trabalho, OntoMate é um motor de busca dirigido a ontologia para auxiliar na mineração de textos na base de dados de genomas Rat. Os componentes que compõe a ferramenta são: coleta de dados, base de dados de artigos, extração de informação e recuperação de informação. A aplicação é armazenada no NoSQL Hadoop/HBase.

Com o mesmo intuito, [Niekler et al. 2014] apresenta seu algoritmo "Leipzig Corpus Miner". O objetivo é lidar com grandes coleções de documentos fazendo uso de algoritmos de mineração de texto para posterior análise de conteúdo. Sua arquitetura armazena os dados em MongoDB devido à flexibilidade para deletar e adicionar registros.

Dentre as principais características dos bancos de dados NoSQL, estão ser distribuída e flexível. Este é ponto explorado pelas arquiteturas que o utilizam para arma-

zenamento e gerenciamento dos dados. Como o formato de texto é aplicável a qualquer modelo NoSQL, pode-se escolher a solução de acordo com a necessidade.

4. Mineração em documentos

Com relação a mineração em documentos, serão considerados aqueles trabalhos que utilizam um formato semi-estruturado como XML ou JSON e que utilizem direta ou indiretamente uma base de dados NoSQL orientada a documentos.

Uma alternativa para se trabalhar com dados semi-estruturados, é convertê-lo a uma estrutura formal. Este processo ocorre através de técnicas de mineração para extração de informação e processamento de linguagem natural. Conforme [Rusu et al. 2013] o intuito é de adicionar estrutura gerando uma saída XML para realizar análises posteriores com as ferramentas existentes. Sua abordagem apresenta as seguintes etapas:

Dado não estruturado → Extração de informação → Análise sintática e semântica → Classificação dos dados → Regras de inferência → Representação em uma estrutura mais formal (XML ou Relações dos dados).

De outra forma o aspecto de documentos pode ser utilizado para analisar dados em redes sociais ou auxiliando nas técnicas relacionadas à mineração.

4.1. Abordagens para analisar dados de redes sociais

É indiscutível o crescimento das redes sociais nos últimos anos. Muitas das soluções NoSQL surgiram devido à demanda gerada por este aumento, como por exemplo Cassandra desenvolvida pelo Facebook, FlockDB criado por um projeto do Twitter usado para análise de gráficos sociais, dentre outros.

Algumas dessas plataformas permitem extrair dados em formatos de marcação de texto. Por exemplo, o Twitter possui uma API (*Application Programming Interface*) para recuperar dados no formato JSON. A partir da extração destes é que são desenvolvidos os trabalhos de [Mansmann et al. 2014] e [Tugores and Colet 2014].

O primeiro mapeia as informações extraídas para XML e a armazena em uma base de dados XML. Então procura extrair cubos multidimensionais através da identificação de partes do conjunto, transformando-os em fatos e dimensões para futuramente aplicar técnicas de OLAP.

De forma semelhante, o segundo armazena os dados em uma base de dados distribuída MongoDB em JSON. No intuito de melhorar o desempenho, as consultas são realizadas com um *plugin* para Python gerando como resultado um mapa com a geolocalização dos tweets nas cidades de Barcelona e Londres.

Este é um dos campos de maior aplicação do NoSQL. Com relação a ferramentas para grande desempenho de análise dos dados de redes sociais e que envolvem diretamente inteligência de negócio a literatura apresenta muitos resultados, porém este não é o foco do trabalho. O objetivo é pesquisar dados semi-estruturados armazenados em uma base de dados NoSQL com possível aplicação de técnicas que envolvam mineração de dados.

4.2. Abordagem em técnicas diretamente relacionadas a mineração

Outra maneira de se trabalhar com NoSQL é aplicando tarefas e técnicas no contexto de mineração, por vezes em conjunto com outras para auxiliar no processo de descoberta de conhecimento.

Uma das técnicas possíveis é a de mineração de tópicos e termos [Lomotey and Deters 2014b]. Sua aplicação a utiliza em conjunto com *clustering* como implementação do *framework* AaaS (*Analysis as a Service*). Esta é testada e implementada em uma base distribuída CouchDB de 30 clientes através do seguinte processo:

Extração de dados do tópico → Organização do termo → Classificação do termo → Clustering

Outra abordagem encontrada utiliza a ferramenta R e um método de agrupamento baseado em regras para melhorar o desempenho. [Kim and Huh 2014] as aplica a um sistema de *logs* personalizados armazenados em uma base de dados MongoDB distribuída. Em sua arquitetura um pré-processador executa a mineração, otimização de dados e após um analisador verifica se a resposta é válida.

De outro modo, [Chai et al. 2013] apresenta uma abordagem de *data warehouse* baseada em documentos para mineração de dados. Neste sistema o processo de ETL (*Extract, Transform e Load*) é carregado a partir de bases de dados heterogêneas através de MapReduce. A seguir os dados são transformados em objetos JSON que são carregados em *clusters* MongoDB.

Pelo fato de bancos de dados NoSQL orientados a documento serem mais populares, este aspecto foi o que mais apresentou resultados. Ao possuir uma estrutura mesmo informal como em linguagens de marcação, as opções de aplicação em técnicas já existentes se tornam maiores.

5. Mineração em grafos

Após a adoção de redes sociais e a proliferação dessas mídias, as pesquisas em descoberta de conhecimento para dados em grafo apresentaram uma revitalização. Ao se trabalhar com grafos, geralmente se está mais preocupado com os relacionamentos e ligações do que com os dados propriamente ditos. Estes modelos são formados por nós, linhas, rótulos e atributos. Seu valor se dá conforme [Begoli 2012] pela facilidade de uso e alto desempenho no acesso a dados de forma associativa e aleatória.

5.1. Abordagens baseadas em técnicas diretamente relacionadas a mineração

Um campo novo a ser explorado é a aplicação de técnicas de mineração, como por exemplo, as que envolvem extração [Lomotey and Deters 2013] e *data warehouse* [Liu and Vitolo 2013].

Em sua ferramenta denominada TouchR [Lomotey and Deters 2013] aplica a metodologia de grafos para melhorar o desempenho na extração dos dados. É implementada com o algoritmo de *Hidden Markov Model* (HMM) e suporta bases de dados como *CouchDB*, *MondoDB*, *Neo4J*, *Cloudata* e *DynamoDB*.

De forma mais diretamente relacionada está a proposta do trabalho em progresso de [Liu and Vitolo 2013], que propõe um conceito de *Graph Data Warehouse*. Seu mo-

dele se baseia no princípio de GCUBE, ou seja, objetiva transformar tabelas e grafos em fatos e dimensões em um cubo multidimensional.

Outro ponto a se destacar é sugestão de uma API para consulta em grafos com as bibliotecas GDML (*Graph Data Manipulation Language*), GDDL (*Graph Data Definition Language*), GML (*Graph Mining Language*) e GSSL (*Graph Structure Statistics Language*).

Poucos trabalhados relacionados à mineração de grafos em NoSQL foram encontrados, porém baseando-se nas ideias iniciais encontradas, outras tarefas e métodos existentes semelhantes aos descritos podem ser aplicados. Acredita-se que a maior contribuição deste modelo esteja relacionada a dados de redes sociais ou redes de relacionamentos em geral.

5.2. Abordagens baseadas na estrutura de grafos

Map Reduce é um *framework* para processar grandes volumes de dados em paralelo implementado por algumas das soluções NoSQL e no qual é baseada a proposta de [Low et al. 2014]. Esta seção trata de projetos para algoritmos de grafos que sejam direta ou indiretamente relacionados ao modelo armazenado em um banco de dados NoSQL orientado a grafos.

Conforme mencionado, [Low et al. 2014] apresenta GraphLab, um *framework* para construção de algoritmos de aprendizagem de máquina paralelos. Seu modelo de dados é baseado em grafos que representam dados e as dependências computacionais.

Outra solução é a de [Gadepally et al. 2015] que procura executar algoritmos de grafo diretamente em bancos de dados NoSQL como Apache Accumulo ou SciDB, que possuem um sistema de armazenamento de dados esparsos. Graphulo realiza o mapeamento entre grafos e álgebra linear, geralmente representando os grafos através de matrizes esparsas ou associativas.

Para concluir a mineração envolve aplicações de diversas áreas. Sua aplicação em dados de grafos se faz necessária, pois auxilia encontrar padrões, detectar anomalias, como por exemplo, para segurança e análise de redes sociais e de sentimentos, dentre outros.

6. Discussões

O estudo aborda uma pesquisa em trabalhos relacionados à mineração de dados semi-estruturados (aqueles representados por estruturas semelhantes a JSON e XML), e não estruturadas delimitadas aos tipos de dados que suportam armazenamento em NoSQL. Para efeitos de pesquisa foram considerados os formatos: texto, documentos e o modelo de grafos.

Há de se considerar que é uma área recente, pois aproximadamente nos últimos 5 anos que se iniciaram os esforços em preencher esta lacuna em análise de dados trazida pelas bases NoSQL. Este fato reflete, por exemplo, o pequeno número de trabalhos relevantes levantados na base da IEEE Xplore, o que indica novas questões para a pesquisa científica.

A maior contribuição em termos de aplicação com relação ao NoSQL diz respeito aos orientados a documentos. Possivelmente isto ocorre devido ao fato de trabalharem

com dados que apesar de não serem formais, permitem uma flexibilidade e maior usabilidade. Como exemplo, no caso do XML e JSON que servem como dados de entrada em diversas ferramentas, inclusive para minerar dados.

Dos trabalhos relacionados com o aspecto de mineração em documentos pode-se observar um comparativo da Tabela 1. Esta aponta uma sumarização das abordagens utilizadas em cada trabalho, além do nome e modelo do banco de dados utilizado para validação das propostas.

Tabela 1. Comparativo dos trabalhos no aspecto mineração em documentos

| Abordagem utilizada | | Modelo do banco de dados | Banco de dados utilizado |
|------------------------|----------------------|--------------------------|--------------------------|
| Cubo multidimensional | OLAP | Centralizado | Base de dados XML |
| Extração de informação | Consultas Python | Distribuído | MongoDB |
| Tópicos e termos | Clustering | Distribuído | CouchDB |
| Método agrupamento | Regras de associação | Distribuído | MongoDB |
| Processo ETL | MapReduce | Distribuído | MongoDB |

Sua área de aplicação é bem ampla, como relatado em redes de dados em tempo real, geolocalização, *logs* personalizados, bases de dados biológicos, além da mineração de opinião e demais informações envolvendo redes sociais como o Twitter.

Dentre as técnicas mais abrangentes na pesquisa estão: cubos de dados multidimensionais e OLAP. Além dessas, pode-se enumerar extração de informação, tópicos, termos, processamento de linguagem natural, etc. Geralmente são aplicadas na construção de *frameworks* bem como em algoritmos próprios.

Por fim observa-se que o formato de texto está mais relacionado com mineração de opinião ou ligado à descoberta de conhecimento em grandes bases de dados já existentes. No que diz respeito a grafos é no geral utilizado para melhor desempenho, seja no acesso ou recuperação de informação, além de sua aplicação tradicional para dados de redes sociais visto que seu modelo se inclina para relacionamentos.

Dessa forma, os maiores esforços concentram-se em trabalhar com dados semi-estruturados, pela diversidades de técnicas já existentes para este tipo. Ainda há possibilidade de se adicionar semi-estrutura a dados sem estrutura alguma. Acredita-se que este seja um dos caminhos para que se possa aproveitar da literatura existente com relação a mineração em XML, por exemplo.

7. Conclusão

O presente trabalho apresentou uma pesquisa em questões de mineração de dados para dados semi-estruturados e não estruturados limitados ao escopo do formato de armazenamento dos bancos de dados NoSQL. Estes ao surgirem e se popularizarem na última década, trouxeram uma lacuna na análise de dados que geralmente trata apenas de dados no formato relacional.

Devido a ser uma área recente, os trabalhos encontrados em sua maioria são pertinentes a dados semi-estruturados como XML e JSON que são os formatos suportados pelas bases de dados NoSQL orientadas a documento. A principal contribuição do trabalho se dá pelo fato das pesquisas apontarem um caminho em direção a esse tipo de dados, ou da conversão de dados não estruturados em semi-estruturados para posterior aplicação de técnicas que envolvem análise.

Também aponta para trabalhos futuros relacionados a descoberta de conhecimento em bancos de dados NoSQL, ou ainda, exploração das técnicas mencionadas como tópicos, termos, extração de informação, cubo multidimensional, etc. Com relação a grafos pode-se identificar pesquisas futuras para análise de dados em redes sociais.

8. Agradecimentos

Este trabalho é financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) através de bolsa de Mestrado do Programa de Pós Graduação em Informática na linha de pesquisa de Linguagens de Programação e Banco de Dados.

Referências

- Begoli, E. (2012). A short survey on the state of the art in architectures and platforms for large scale data analysis and knowledge discovery from data. In Proceedings of the WICSA/ECSA 2012 Companion Volume, pages 177–183. ACM.
- Chai, H., Wu, G., and Zhao, Y. (2013). A document-based data warehousing approach for large scale data mining. In Proceedings of the 2012 International Conference on Pervasive Computing and the Networked World, ICPCA/SWS'12, pages 69–81, Berlin, Heidelberg. Springer-Verlag.
- Chakraborty, G. (2014). Analysis of unstructured data: Applications of text analytics and sentiment mining. In SAS Global Forum. Washington, DC, pages 1288–2014.
- Gadeppally, V., Bolewski, J., Hook, D., Hutchison, D., Miller, B., and Kepner, J. (2015). Graphulo: Linear algebra graph kernels for nosql databases. In Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2015 IEEE International, pages 822–830. IEEE.
- Han, J., Kamber, M., and Pei, J. (2011). Data mining: concepts and techniques: concepts and techniques. Elsevier.
- Kanimozhi, K. and Venkatesan, M. (2015). Unstructured data analysis- a survey. International Journal of Advanced Research in Computer and Communication Engineering.
- Kim, J. S., Yang, M. H., Hwang, Y. J., Jeon, S. H., Kim, K., Jung, I., Choi, C.-H., Cho, W.-S., and Na, J. (2012). Customer preference analysis based on sns data. In Cloud and Green Computing (CGC), 2012 Second International Conference on, pages 609–613. IEEE.
- Kim, Y.-H. and Huh, E.-N. (2014). A rule-based data grouping method for personalized log analysis system in big data computing. In Innovative Computing Technology (INTECH), 2014 Fourth International Conference on, pages 109–114. IEEE.

- Liu, W., Laulederkind, S. J., Hayman, G. T., Wang, S.-J., Nigam, R., Smith, J. R., De Pons, J., Dwinell, M. R., and Shimoyama, M. (2015). Ontomate: a text-mining tool aiding curation at the rat genome database. *Database*, 2015:bau129.
- Liu, Y. and Vitolo, T. M. (2013). Graph data warehouse: Steps to integrating graph databases into the traditional conceptual structure of a data warehouse. In *Big Data (BigData Congress), 2013 IEEE International Congress on*, pages 433–434. IEEE.
- Lomotey, R. K. and Deters, R. (2013). Terms extraction from unstructured data silos. In *System of Systems Engineering (SoSE), 2013 8th International Conference on*, pages 19–24. IEEE.
- Lomotey, R. K. and Deters, R. (2014a). Data mining from nosql document-append style storages. In *Web Services (ICWS), 2014 IEEE International Conference on*, pages 385–392. IEEE.
- Lomotey, R. K. and Deters, R. (2014b). Terms mining in document-based nosql: Response to unstructured data. In *Big Data (BigData Congress), 2014 IEEE International Congress on*, pages 661–668. IEEE.
- Lomotey, R. K. and Deters, R. (2014c). Towards knowledge discovery in big data. In *Service Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on*, pages 181–191. IEEE.
- Low, Y., Gonzalez, J. E., Kyrola, A., Bickson, D., Guestrin, C. E., and Hellerstein, J. (2014). Graphlab: A new framework for parallel machine learning. *arXiv preprint arXiv:1408.2041*.
- Mansmann, S., Rehman, N. U., Weiler, A., and Scholl, M. H. (2014). Discovering olap dimensions in semi-structured data. *Information Systems*, 44:120–133.
- McKendrick, J. (2011). The post-relational reality sets in: 2011 survey on unstructured data. *Unisphere Research*.
- Niekler, A., Wiedemann, G., and Heyer, G. (2014). Leipzig corpus miner-a text mining infrastructure for qualitative data analysis. In *Terminology and Knowledge Engineering 2014*, pages 10–p.
- Padhy, R. P., Patra, M. R., and Satapathy, S. C. (2011). Rdbms to nosql: Reviewing some next-generation non-relational databases. *International Journal of Advanced Engineering Science and Technologies*, 11(1):15–30.
- Rusu, O., Halcu, I., Grigoriu, O., Neculoiu, G., Sandulescu, V., Marinescu, M., and Marinescu, V. (2013). Converting unstructured and semi-structured data into knowledge. In *Roedunet International Conference (RoEduNet), 2013 11th*, pages 1–4. IEEE.
- Sadalage, P. J. and Fowler, M. (2012). *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education.
- Tugores, A. and Colet, P. (2014). Mining online social networks with python to study urban mobility. *arXiv preprint arXiv:1404.6966*.
- Wylie, B., Dunlavy, D., Davis IV, W., and Baumes, J. (2012). Using nosql databases for streaming network analysis. In *Large Data Analysis and Visualization (LDAV), 2012 IEEE Symposium on*, pages 121–124. IEEE.

aper:152907_1

Mineração de opiniões em microblogs com abordagem CESA

Alex M. G. de Almeida¹, Sylvio Barbon Jr.¹, Rodrigo A. Igawa¹, Stella Naomi Moriguchi²

¹Universidade Estadual de Londrina (UEL)
Caixa Postal 10.011 – 86.057-970 – Londrina – PR – Brazil

²Universidade Federal de Uberlândia (UFU)
Uberlândia – MG.

Abstract. *The recent increased use of digital social media in the company's every day raised interest in the study of techniques to obtain knowledge based on text messages. This work describes the Opinion Mining in microblogs based on CESA (Crowd Explicit Sentiment Analysis) applied in a data set formed by followers posts of True Blood series. Rating Feeling is performed by a succession of Latent Semantic Analysis and projection of sentences on the polarized index of terms. Experiment results show an overall efficiency about 81% reaching 86% for positive polarity words.*

Resumo. *O crescente uso das mídias sociais digitais no cotidiano da sociedade tem estimulado o estudo de técnicas para obtenção de conhecimento baseado em postagens de texto. Este trabalho contempla a Mineração de Opiniões em microblogs por meio da CESA (Crowd Explicit Sentiment Analysis) aplicada em uma base de postagens dos fãs do seriado “True Blood”. A Classificação de Sentimento é realizada pela sucessão da Análise Semântica Latente e da projeção de sentenças em índices de termos polarizados. Os resultados obtidos no experimento mostraram eficiência geral de 81% alcançando 86% para palavras polarizadas positivamente.*

1. Introdução

Análise de Sentimento ou Mineração de Opiniões é o campo de estudo que analisa opiniões, sentimentos, atitudes e emoções dos indivíduos por meio da linguagem escrita. Atualmente a Mineração de Opinião destaca-se dentro do Processamento de Linguagem Natural como uma das mais ativas linhas de pesquisa, podendo ser associada às técnicas de Recuperação de Informação e Aprendizagem de Máquina, que resultam na classificação de peso semântico sentimental da forma escrita. Ainda que se possa afirmar que emoções são subjetivas ao passo que sentimentos são ausentes de emoção, pode-se compreender a análise de sentimento como esforço computacional de identificar sentimentos, opiniões e avaliações em formato textual [Liu and Zhang 2012].

O crescimento do interesse na Mineração de Opinião coincide com o recente crescimento das mídias sociais digitais, tais como: fóruns de discussões, blogs, microblogs e redes sociais digitais. Assim, foi produzido pela primeira vez na história humana, conteúdo textual com a possibilidade de avaliações diversas sobre variados assuntos [Pak and Paroubek 2010, Agarwal et al. 2011].

As opiniões são pontos centrais de muitas atividades humanas pois funcionam como influenciadores de comportamento e, muitas vezes, antes da tomada de decisão,

procuram-se opiniões de outrem. Empresas e organizações têm grande interesse em conhecer opiniões sobre seus serviços ou produtos de forma a obter vantagens competitivas e avaliar o mercado no qual estão inseridas [Liu 2012].

Em suma, as mídias sociais digitais oferecem um vasto conjunto de dados que possibilitam extrair opiniões de parcela de uma sociedade por meio da linguagem escrita [Bollen et al. 2011].

Neste contexto, pretende-se aplicar uma abordagem de classificação de sentimento em microblogs conhecida como CESA (*Crowd Explicit Sentiment Analysis*). A ideia é "*Let the crowd express itself*", usando o *corpora* para criar um vetor de sentimentos por meio da projeção do vetor de palavras de cada sentença [Montejo-Ráez et al. 2014].

O trabalho está organizado da seguinte forma. Primeiro será apresentado um referencial teórico sobre os trabalhos relacionados ao assunto seguida de uma fundamentação teórica. Na sequência será descrita brevemente a abordagem CESA e metodologia. Na seção seguinte, detalhadamente, discorrer-se-á sobre o experimento e resultados obtidos e, finalmente, sugerir-se-ão proposições de melhorias e trabalhos futuros.

2. Trabalhos relacionados

Classificação de sentimentos é um assunto em destaque dentre as linhas de pesquisa. Deve-se ter clareza em afirmar que palavras e frases carregam no seu arcabouço sentimentos positivos ou negativos que caracterizam instrumentos de análise de sentimentos. Esta atividade é comumente chamada de classificação de sentimento em nível de documento e assume-se que um opinante (usuário de microblog) expressa sua opinião sobre um determinado assunto. A classificação de sentimento, para polaridade negativa ou positiva, é um problema de classificação de duas classes. Além disso trata-se também de um problema de classificação de texto[Liu and Zhang 2012].

Ainda se tratando de um problema de classificação de texto, como tal, qualquer método de aprendizado supervisionado pode ser aplicado, tais como Naive Bayes e Máquina de Vetor de Suporte [Prabowo and Thelwall 2009]. O ponto chave da classificação de sentimentos é a engenhosidade da combinação de técnicas que possibilitam a classificação, tais como *stemming*, remoção de *stopwords*, lemantização, TF-IDF (*term frequency-inverse document frequency*), n-GRAMSs, POS tags (*part-of-speech*) [Manning et al. 2008] e listas de palavras sentimentais (*lexicons*) [Potts 2011].

Lexicon é um vetor composto por palavras semanticamente agrupadas em classes. A estrutura tradicional de um *lexicon* aproxima-se à de um dicionário com uma definição para cada palavra [Turney 2002]. Para a análise de sentimento o *lexicon* é o conjunto de palavras com peso semântico sentimental associadas a informações de polaridade.

A classificação de sentimento com aprendizado supervisionado foi apresentada em 2003 e 2004 realizada com base no *feedback* de clientes por Nasukawa e Yi [Nasukawa and Yi 2003] e Gamon [Gamon 2004]. Ainda em 2004, Pang e Lee [Pang and Lee 2005] aplicaram algoritmo de cortes mínimos em grafos para auxiliar na tarefa de classificação e Mullen e Colliers [Mullen and Collier 2004] utilizaram análise sintática unida às técnicas tradicionais. Em 2006 Ng et al. [Ng et al. 2006] apresentaram uso da linguística; em 2008 Abbasi et al. [Abbasi et al. 2008] propuseram a utilização de Algoritmos Genéticos para classificação de sentimento em diferentes lin-

guagens; em 2006 Ding, Chris, et al. [Ding et al. 2006] fizeram uso de uma matriz não negativa ortogonal e entre 2009 e 2010 Dasgupta e Ng [Dasgupta and Ng 2009] e Li et al. [Li et al. 2010] realizaram experimentos com uma abordagem de aprendizado semi-supervisionada. Ainda em 2009 Martineau e Finin [Martineau and Finin 2009] elaboraram o DELTA TF-IDF, mais recentemente em 2011 Bespalov et al. [Bespakov et al. 2011] usaram método de classificação latente com n-gramas.

Da abordagem de classificação não supervisionada pode-se mencionar o trabalho de Turney [Turney 2002] cuja classificação é baseada em padrões sintáticos - por meio de *part of speech*- que são frequentemente usados para expressar opiniões [Pang and Lee 2008]. Os trabalhos citados oferecem contribuições para classificação de sentimentos em microblogs, que têm sido fonte de dados de expressão de opiniões humanas pós massificação das mídias sociais.

Para este trabalho foi adotada a abordagem CESA, classificação supervisionada, pelo fato de necessitar de poucas mudanças para o uso em outros idiomas que não o inglês, produzindo ainda bons resultados [Montejo-Ráez et al. 2014].

3. O seriado “*True Blood*”

True Blood é uma das séries de maior sucesso do canal de televisão norte-americano Home Box Office – HBO que recebeu perto de 30 premiações diversas e 115 indicações entre os anos de 2009 e 2014. Tratando da coexistência de vampiros e humanos em Bon Temps, uma pequena cidade fictícia localizada na Louisiana, o seriado foca Sookie Stackhouse, uma garçonete telepata que se apaixona pelo vampiro Bill Compton. Em 2014, pode-se perceber uma mudança nas postagens devido ao final da série, além das manifestações de admiração ou de amor, houve manifestações de tristeza explícitas.

4. Abordagem CESA

A proposição do uso de uma coleção de documentos para formar índices de novos documentos de Gabrilovich, E., & Markovitch [Gabrilovich and Markovitch 2007], ESA - *Explicit Semantic Analysis* - visa representar o significado de textos numa matriz de conceitos derivados de uma fonte de dados textual ou *corpora*. Por esta razão a abordagem CESA usa o *corpora* com a finalidade de formar um vetor de sentimentos.

Para formação do *lexicon* v_s foi utilizada a base de termos polarizada WeFeel-Fine¹ [Kamvar and Harris 2011], que é um website que coleta de diversas mídias sociais, milhões de sentenças contendo “*I feel*” ou “*I am feeling*”, formando uma base de termos sentimentais, atualmente contém 2178 termos. O WeFeelFine é um recurso valioso para Mineração de Opinião, em particular para trabalhos baseados em mídias sociais, pela constante atualização de sua base [Montejo-Ráez et al. 2014].

O processo CESA é ilustrado na Figura 1, inicia-se na aquisição textual da fonte de dados e na sequência, filtragem. Esta filtragem realiza as tarefas de pré-processamento que resultarão no vetor de sentimentos - a ser polarizado manualmente - e no corpus com seus respectivos TF-IDFs. A formação da MATRIZ M_{mn} , ponto chave da abordagem CESA, é resultado da combinação do vetor de sentimentos v_s e do corpus, onde m é o tamanho do corpus e n o número de sentimentos obtidos na filtragem. O processo CESA

¹<http://wefelfine.org>

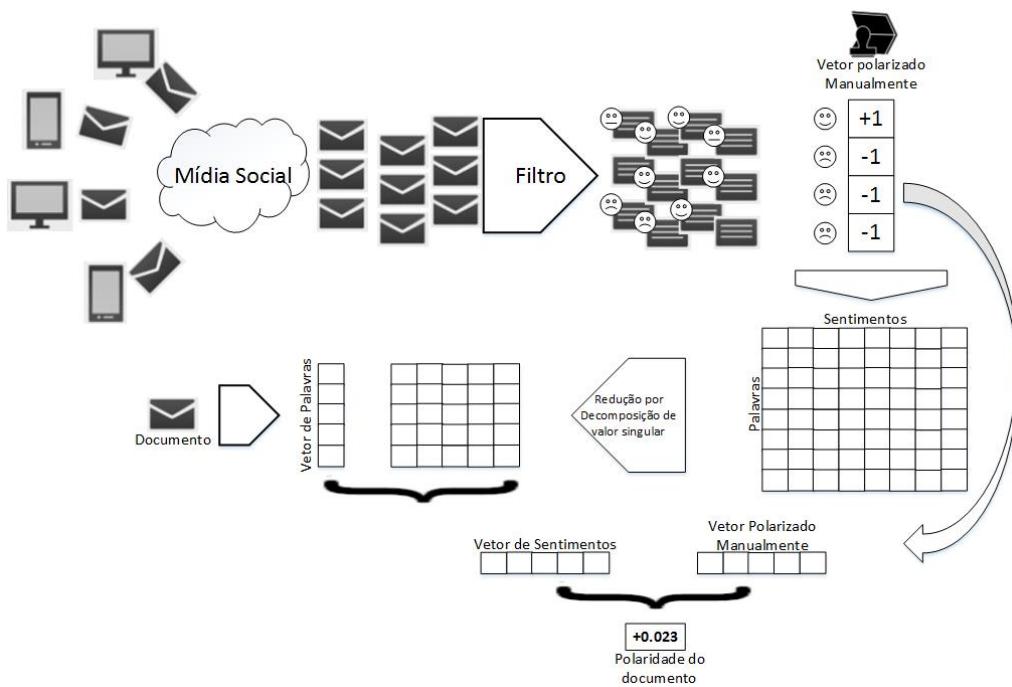


Figura 1. Diagrama CESA. Adaptado de [Montejo-Ráez et al. 2014]

de obtenção de polaridade de um dado documento é realizado através da consecução das seguintes tarefas:

1. Transformar a matriz M , por decomposição de valores singulares preservando 95% da variança obtidos da decomposição dos autovalores não nulos, numa matriz M^* .
2. Do documento d a classificar, obter o vetor de palavras v_d com respectivos TF-IDF.
3. Obter o vetor x de sentimentos por $v_d^* M^*$.
4. Por fim a polaridade final, P , obtida com:

$$P(d) = \frac{1}{|F|} \sum x_l \cdot f_l$$

onde:

- $P(d)$ - polaridade do documento;
- F - conjunto de sentimentos;
- x_l - peso do sentimento no vetor de sentimento;
- f_l - polaridade do sentimento [-1, 1];

Do resultado obtido de $P(d)$, tem-se -1 para documentos com polaridade negativa e 1 para documentos com polaridade positiva.

O procedimento apresentado na Figura 1 difere de [Montejo-Ráez et al. 2014] pela supressão da filtragem sintática (POS tags) porque neste trabalho não houve integração com o dicionário SenticNet [Cambria et al. 2014] e os resultados obtidos por [Montejo-Ráez et al. 2014] com o uso WeFeelFine corroboram com esta decisão. A opção de não utilizar o SenticNet deu-se porque pretende-se para trabalhos futuros desenvolver metodologias de mineração de opiniões para usuário lusófonos.

5. Metodologia

A Mineração de Opiniões é um subconjunto da Mineração de Dados. Como tal é factível realizar as etapas: Aquisição de Dados, Extração de Características e Classificação [Harb et al. 2008].

A etapa de Aquisição de Dados utilizou como fonte o serviço de microlog Twitter. Extraíram-se do microblog dados referentes aos seguidores do seriado “*True Blood*”, coletados entre os dias 15/06/2014 e 24/06/2015, totalizando um conjunto de 19.077 tweets e um vocabulário de 20.926 palavras distintas. Foi escolhido um subconjunto de *tweets* dos seguidores do seriado True Blood por conta da expressividade de tensões e afinidades de inúmeros leitores de opiniões divergentes. Esta escolha foi feita também pelo fato de se tratar de uma narrativa contemporânea sobre diferentes grupos sociais e dos julgamentos morais gerados pelas relações sociais do seriado [Panse and Rothermel 2014, Hardy 2011].

A redução e limpeza, contidas na etapa de Extração de Características, deu-se seguindo as seguintes tarefas de pré-processamento:

1. *Tokenizing*: Dada uma sequência de caracteres, *tokenization* é a tarefa de separar em partes, chamadas *tokens*, podendo simultaneamente separar certos caracteres, como por exemplo pontuações [Manning et al. 2008].
2. *Stemming*: procedimento computacional que reduz palavras com a mesma raiz a uma única forma, usualmente realizada por meio da remoção de afixos [Lovins 1968].
3. Remoção de *stopwords*: Palavras que aparecem com muita frequência em textos e sentenças, tais como artigos e preposições, auxiliam na construção de ideias. Entretanto são desprovidas de peso semântico são removidas antes de classificar os termos [Manning et al. 2008].
4. Substituição de palavras especiais por marcadores
 - (a) menções (ou citações) por MENTION
 - (b) *tags* html por HTML
 - (c) *hashtags* por HASHTAG
 - (d) *emoticons* por POSITIVE ou NEGATIVE
5. Para cada elemento de v_s obteve-se o TF-IDF ou DELTA TF-IDF [Martineau and Finin 2009]. Formando assim uma matriz M_{mn} onde m é vocabulário e n o vetor de sentimentos.

Para a classificação foi utilizada uma lista contendo os tweets e suas polaridades esperadas, promovendo a avaliação da abordagem automaticamente. A polaridade esperada foi atribuída manualmente por meio da soma dos valores polarizados dos termos identificados em cada tweet. Conforme a Tabela 1, tomando o tweet - **Sonic sweet tea & tacos from Bill's, breakfast of champions dawg** - tem-se o termo **sweet** com polaridade 1 como único termo polar na sentença e para este caso é atribuído ao tweet a polaridade esperada “1”, logo positiva.

Este procedimento foi realizado para uma amostra de 1663 tweets com objetivo de preservar a proporção entre sentenças neutras, positivas e negativas, tanto na amostra quanto para formação do corpus, conforme apontado na Tabela 2.

Da amostra polarizada manualmente, 141 tweets foram atribuídos com polaridade negativa e aos 1522 restantes atribuiram-se polaridades positiva e neutra.

Tabela 1. Exemplos de amostra com polaridade esperada

| Tweet | Polaridade |
|---|------------|
| Sonic sweet tea tacos from Bill's, breakfast of champions dawg. | 1 |
| RT@FilmLadd: Ever wonder if Hillary just wants to be President so she can cheat on Bill in the oval office? #payback | 0 |
| Tate Modern ; 061514. Vision as reception, vision as reflection, vision as projection. | 1 |
| Almost feel bad for these bill Jill and kelsie strangers but they kinda brought it on themselves http://t.co/IJHEZ6Ze1c | -1 |
| Hey media the tea party's win isn't that Eric Cantor lost . It's that there guy won. Get it right num sculls | -1 |
| Dollar dollar bill yall | 0 |

Tabela 2. Relação base por amostra

| 2*Polaridade | Corpus | | Amostra | |
|--------------|------------|-------|------------|-------|
| | Quantidade | % | Quantidade | % |
| Negativas | 1821 | 9,55 | 141 | 8,48 |
| Positivas | 4304 | 22,56 | 427 | 25,68 |
| Neutras | 12952 | 67,89 | 1095 | 65,84 |

6. Resultados e Discussão

Após etapa de aquisição e pré-processamento, foram obtidos os resultados de classificação. Os termos contidos no *corpus* e presentes na base WeFeelFine totalizaram em um *lexicon* com 490 termos polarizados, divididos em 251 termos positivos e 239 termos negativos, conforme a Etapa 3 da Seção 3. A Tabela 3 exibe alguns dos termos polarizados e seus respectivos valores de TF-IDFs calculados pela equação 1, que possibilitaram a formação da matriz M e M^* .

$$\text{TF-IDF}_{t,d} = (\text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t} \quad (1)$$

A sumarização dos resultados está presente na Tabela 5, organizada com as quantidades de acertos e erros para os tweets positivos e negativos. Foi possível calcular os índices de acertos e como resultado principal obteve-se acurácia de 81%, número elevado considerando, por exemplo, o trabalho de Liu [Liu 2012].

Tabela 3. Lexicon com polaridade e TF-IDF das palavras mais e menos frequentes

| Palavra | Polaridade | TF-IDF |
|---------------|------------|--------|
| bad | -1 | 6,75 |
| best | 1 | 6,71 |
| better | 1 | 6,70 |
| dead | -1 | 6,79 |
| ecstatic | -1 | 0,00 |
| irresponsible | -1 | 0,00 |
| naive | -1 | 0,00 |
| rough | -1 | 0,00 |

Entretanto, os resultados apresentam um índice de acertos discrepantes entre as sentenças positivas e negativas, 86% e 38% respectivamente. Este fato está associado à desproporcionalidade entre a quantidade de sentenças negativas e positivas (Tabela 2), dado que a amostra possui quantidade de sentenças positivas 11 vezes maior do que a de sentenças negativas; somado a alta precisão para sentenças positivas corrobora para um resultado final aceitável mesmo com uma baixa precisão para sentenças negativas.

Ainda referente à ineficiência da classificação das sentenças negativas, identificou-se que parte significativa dos termos com TF-IDF igual a “0” eram de polaridade negativa, em decorrência de seu TF (*Term Frequence*) igual a “1”, produzindo consequentemente uma matriz M_{mn} com valores zerados para as colunas n (referentes aos *lexicons*), constatou-se ainda que as sentenças contendo os referidos termos invariavelmente produziram erro de classificação para os termos negativos e positivos, como exibido na Tabela 4.

Tabela 4. Relação erros por TF-IDF

| Polaridade | TF-IDF>0 | | TF-IDF=0 | |
|------------|------------|----|------------|----|
| | Quantidade | % | Quantidade | % |
| Negativas | 48 | 54 | 40 | 46 |
| Positivas | 199 | 90 | 21 | 10 |

Assumindo a hipótese de processar uma amostra com quantidades de sentenças positivas e negativas na mesma proporção, seria produzida uma acurácia final na ordem de 61%, semelhante aos resultados obtidos em [Montejo-Rález et al. 2014] entre 37% e 72% em média, acumulando todas as polaridades.

Tabela 5. Resultado do Experimento

| TWEETS | ACERTOS | ERROS | EFICÊNCIA |
|-----------|---------|-------|-----------|
| POSITIVOS | 1302 | 220 | 86% |
| NEGATIVOS | 53 | 88 | 38% |
| TOTAL | 1355 | 308 | 81% |

No que tange aos usuários do Twiter do True Blood, espaço amostral deste trabalho, considerando que a análise de sentimento restringiu-se simplesmente a polaridade

das sentenças, ou seja, se uma sentença carrega consigo sentimentos positivos ou negativos, desconsiderando personagem, fatos ou mesmo produtos pode-se verificar que o sentimento geral dos usuários é positivo em relação ao seriado; corroborados pelos resultados.

7. Considerações finais

Este trabalho apresentou o uso da abordagem CESA para classificação de polaridade para microblogs, acompanhado com o uso da base WeFeelFine como fonte para formação do *lexicon*. Diferenciando da abordagem original na classificação sintática dos termos, justificado pelo fato de não utilizar o SenticNet.

Para executar o experimento utilizou-se uma base de dados de postagens do Twitter sobre o seriado True Blood, realizando o processo de polarização manual e subsequente classificação pelo processo CESA. Os resultados do experimentos apresentaram um índice de acerto para sentenças neutras e positivas, total de 86%, superior ao da literatura. Entretanto para sentenças negativas o resultado foi similar ao estado da arte, com 38%. Ainda que o resultado aferido nas sentenças negativas pudesse comprometer o experimento, a proporcionalidade entre sentenças positivas e negativas pesou favoravelmente no resultado geral.

A característica de ineficiência de classificação das sentenças negativas pode, em trabalhos futuros, ser melhor investigada quando da formação dos vetores de sentimentos, produto de determinação dos auto-valores da matriz de termos e sentimentos (produto do TF-IDF pela polaridade). Esta mudança pode modificar a geração elementos de valores nulos, provável cerne do problema de polaridade negativa.

Outro trabalho a ser realizado está vinculado a possibilidade da utilização do CESA para conteúdo em língua portuguesa, desta forma espera-se desenvolver metodologias para mineração de opiniões de usuários brasileiros.

Pretende-se também explorar classificação de polaridade em conjunto com detecção de sarcasmo que podem contribuir na determinação de falsos positivos e negativos.

Referências

- Abbasi, A., Chen, H., and Salem, A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems (TOIS)*, 26(3):12.
- Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, pages 30–38. Association for Computational Linguistics.
- Bespakov, D., Bai, B., Qi, Y., and Shokoufandeh, A. (2011). Sentiment classification based on supervised latent n-gram analysis. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 375–382. ACM.
- Bollen, J., Mao, H., and Pepe, A. (2011). Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *ICWSM*.

- Cambria, E., Olsher, D., and Rajagopal, D. (2014). Senticnet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. In *Twenty-eighth AAAI conference on artificial intelligence*.
- Dasgupta, S. and Ng, V. (2009). Mine the easy, classify the hard: a semi-supervised approach to automatic sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 701–709. Association for Computational Linguistics.
- Ding, C., Li, T., Peng, W., and Park, H. (2006). Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM.
- Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611.
- Gamon, M. (2004). Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th international conference on Computational Linguistics*, page 841. Association for Computational Linguistics.
- Harb, A., Planté, M., Dray, G., Roche, M., Trouset, F., and Poncelet, P. (2008). Web opinion mining: How to extract opinions from blogs? In *Proceedings of the 5th international conference on Soft computing as transdisciplinary science and technology*, pages 211–217. ACM.
- Hardy, J. (2011). Mapping commercial intertextuality: Hbo’s true blood. *Convergence: The International Journal of Research into New Media Technologies*, 17(1):7–17.
- Kamvar, S. D. and Harris, J. (2011). We feel fine and searching the emotional web. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 117–126. ACM.
- Li, S., Huang, C.-R., Zhou, G., and Lee, S. Y. M. (2010). Employing personal/impersonal views in supervised and semi-supervised sentiment classification. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 414–423. Association for Computational Linguistics.
- Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.
- Liu, B. and Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In *Mining Text Data*, pages 415–463. Springer.
- Lovins, J. B. (1968). *Development of a stemming algorithm*. MIT Information Processing Group, Electronic Systems Laboratory.
- Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.
- Martineau, J. and Finin, T. (2009). Delta tfidf: An improved feature space for sentiment analysis. In *ICWSM*.
- Montejo-Ráez, A., Díaz-Galiano, M., and Ureña-López, L. (2014). Crowd explicit sentiment analysis. *Knowledge-Based Systems*.

- Mullen, T. and Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources. In *EMNLP*, volume 4, pages 412–418.
- Nasukawa, T. and Yi, J. (2003). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM.
- Ng, V., Dasgupta, S., and Arifin, S. (2006). Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 611–618. Association for Computational Linguistics.
- Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*.
- Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 115–124. Association for Computational Linguistics.
- Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.
- Panse, S. and Rothermel, D. (2014). *A Critique of Judgment in Film and Television*. Palgrave Macmillan.
- Potts, C. (2011). Sentiment symposium tutorial. In *Sentiment Symposium Tutorial. Acknowledgments*.
- Prabowo, R. and Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2):143–157.
- Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics.

aper:152877_1

Um processo de avaliação de dados em um Data Warehouse

Tania M. Cernach¹, Edit Grassiani¹, Renata M. de Oliveira², Carlos H. Arima²

¹IPT – Instituto de Pesquisas Tecnológicas do Estado de São Paulo – SP – Brasil

²CEETEPS – Centro Paula Souza, São Paulo – SP – Brasil

tania.antunes@yahoo.com.br, edit.grassiani@gmail.com,
renata_mno@hotmail.com, charima@uol.com.br

Abstract: This paper describes a case study that evaluates a data sample in its current stage, in real systems that deeply depend on reliable data inside a financial institution. This way, a process is proposed here for the data quality assessment based on methods and processes surveyed in literature. The definition of a data sample, dimensions, quality metrics and relevant rules to the business context, as well as data measurement and analysis of the results were performed here. After the evaluation it is possible to draw action plans, aiming the continuous improvement of the data. Action plans are not part of this paper.

Resumo: Este trabalho descreve um estudo de caso para efetuar a avaliação do estado atual de uma amostra de dados, contidos em sistemas reais que dependem fundamentalmente de dados confiáveis, dentro de uma instituição financeira. Dessa forma, é proposto um processo para a avaliação da qualidade dos dados tomando como base métodos e processos pesquisados na literatura. A definição de uma amostra de dados, de dimensões, de métricas e de regras de qualidade relevantes ao contexto do negócio, bem como a medição dos dados e análise dos resultados foram executados. Após a avaliação é possível traçar planos de ação, visando a melhoria contínua dos dados. Os planos de ação não fazem parte dessa pesquisa.

1. INTRODUÇÃO

Qualidade de dados (QD) é um tema de estudo ainda a ser bastante explorado por autores da área. Sua aplicação nas organizações sofre processos de amadurecimento.

A qualidade de dados não é só obtida por inspeção e correção dos dados que implicam custos decorrentes da qualidade pobre de dados. A qualidade de dados dentro de uma empresa é resultante de um projeto de qualidade inserido nos seus processos de negócio. Esse projeto de qualidade provê técnicas de qualidade conhecidas como: Planejamento-Execução-Análise e Definir-Medir-Analisar-Melhorar-Controlar os dados da empresa dentro de um contexto de negócio, conforme retrata English (2009).

O objetivo desse artigo é descrever um estudo de caso usando métodos e processos de QD propostos por Storey e Wang (1998), Maydanchik (2007) e English (2009) e propor um processo de avaliação de dados.

O artigo está estruturado da seguinte forma: contextualização da empresa da qual foi retirada a amostra de dados para o estudo; breve descrição dos métodos de QD

pesquisados e que serviram de base para o estudo; a metodologia aplicada, descrevendo cada passo executado para a construção de um processo de avaliação de dados; resultados da avaliação dos dados da amostra e conclusões finais.

O estudo de caso aplicou um processo de avaliação de dados a fim de verificar o estado atual de uma amostra dos dados construído em uma instituição financeira, com base nas dimensões e métricas de qualidade relevantes ao contexto do negócio.

A amostra de dados pertence a uma instituição financeira de caráter privado e que tem como principal objetivo a atividade bancária, oferecendo serviços a um público variado. Entre os serviços estão os produtos de seguros, de previdência, de crédito, gestão de ativos, entre outros. Sua rede de atendimento compreende aproximadamente 5000 pontos de atendimento em todo o Brasil e se expande ao exterior, também, em alguns países da Europa, Ásia, Oriente Médio e Américas.

A empresa reúne seus dados em repositórios por meio de processos de extração, transformação e carga ETL (*extract, transform and load*). Os dados originários são de sistemas transacionais da própria empresa, armazenando-os sob forma de tabelas para posterior consulta. Dentro de um cenário mais específico, essas informações servem como matéria-prima, ou seja, dados brutos para as áreas de análise de risco estudarem a probabilidade de clientes não quitarem seus contratos de crédito adquiridos junto à instituição financeira.

A estratégia adotada pela empresa foi inserir pontos de controle de qualidade nos seus processos de ETL mais críticos. Os pontos de controle são definidos por meio de scripts que efetuam operação de soma, média, frequência de valores para variáveis importantes do contexto do negócio e compararam-se os valores obtidos no mês corrente com os valores obtidos na última análise, normalmente valores de um mês anterior. São atitudes bastante reativas, pois o sistema não impede a entrada de informações consideradas sem qualidade, apenas as identifica e, havendo necessidade, os sistemas fontes ou o próprio sistema de risco financeiro são acionados para análise e correção das informações de forma pontual.

Não há a conscientização de QD com base em atributos de qualidade. Há a noção de faixas de valores permitidas para campos numéricos e conteúdos esperados para campos discretos e, com base nessas regras pré-definidas, os pontos de controle alertam os processos que consomem os dados até a entrega deles aos usuários finais. Ao longo do tempo, sentiu-se a necessidade de verificar como os dados estão em termos de qualidade antes mesmo de utilizá-los.

O que se procura com a pesquisa é uma forma de avaliar os dados ainda nos sistemas geradores dos dados e nos sistemas intermediários que os consomem, antes de entregar os dados para os sistemas finais responsáveis pela geração dos resultados divulgados pela empresa. Busca-se também a garantia de que ao utilizar dados exista uma aceitação mínima de qualidade.

A seguir uma descrição dos métodos e processos de QD que serviram de base para o estudo.

1.1 Métodos de QD

O método TDQM (*Total Data Quality Management*), proposto por Richard Wang e Stuart Madnick (1993) no programa de Gerenciamento Total de Qualidade do MIT, apresenta um ciclo produtivo e consiste das fases: definir, medir, analisar e aprimorar QD continuamente, sendo essencial para prover alta qualidade no produto de informação (PI).

Na fase definir, capturam-se as características de um PI sob dois níveis de percepção: em um nível mais amplo, são identificadas as funcionalidades ou requisitos para os consumidores da informação. Em um segundo nível, mais detalhado, define-se características básicas do PI e seus relacionamentos, que podem ser representados por meio de um modelo ER (entidade-relacionamento).

Na fase medir, o principal é a definição de métricas objetivas e a aplicação dessas métricas às diferentes fontes de dados e às informações da aplicação e em qualquer estágio do seu ciclo de vida.

Na fase analisar, são identificadas as causas-raízes para os problemas de QD correntes. As causas são divididas por papéis: causas relacionadas aos fornecedores, aos mantenedores e aos consumidores. Nessa fase são efetuados também os planos de melhoria envolvendo processos de limpeza de dados e redesenho dos processos.

Por fim, na fase melhorar, o objetivo é priorizar as áreas chaves para o plano de melhoria, com o alinhamento das características chaves do PI, de acordo com as necessidades do negócio. É uma fase de planejamento do plano de melhoria.

Para efetuar o estudo de caso, a avaliação de QD teve como apoio as fases definir e medir da metodologia TDQM.

Outro método TQdM (*Total Quality data Management*) que é atualmente conhecido por TIQM – *Total Information Quality Management*, surgiu em 1992 e foi elaborado por Larry English (2009). Pode ser definido como um sistema prático de princípios de qualidade, de processos e de técnicas aplicados para a medição de QD e melhoria de processos dentro de uma organização, visando a eliminação das causas raízes da qualidade pobre de informações.

O método TIQM tem como objetivo a melhoria contínua da qualidade. Dessa forma, para efetuar a avaliação de uma amostra de dados, esse método serviu como um apoio, principalmente na etapa de medição dos dados que avalia a qualidade da informação, medindo os atributos críticos, dentre eles, completude, precisão, atualidade, relevância, apresentação dos dados, entre outros, sob a visão de consumidores. Outro processo de apoio foi a etapa que mede o quanto correta, clara e completa está a definição de um processo de negócio sob a visão de pessoas que utilizam as informações e provêem dados, afim de estabelecer uma comunicação comum de entendimento durante a realização de suas atividades, pois trata o conteúdo dos dados com base nas especificações contidas nos metadados.

Maydanchik (2007) propõe um projeto de avaliação de qualidade de dados que pode ser dividido em quatro fases:

Fase de Planejamento: nessa fase define-se o escopo de forma bem delimitada e estreita. São conhecidos as tabelas, registros e campos que são relevantes e podem prover claras definições de qualidade de dados.

Fase de Preparação: o objetivo dessa fase é estar pronto para desenvolver regras de qualidade. Envolve carregar os dados em uma área de trabalho, desenvolver catálogos e modelos de dados.

Fase de Implementação: essa é a fase principal de uma avaliação de qualidade de dados. Antes de escrever as regras de qualidade propriamente ditas nessa fase, é necessário efetuar a atividade de *data profiling*. *Profiling* é uma descrição dos dados, como eles se apresentam no momento em que estão sendo avaliados, pois ao longo do tempo, modelos de dados e dicionários se tornam não corretos e não completos. As regras de qualidade correspondem à ferramenta principal de um projeto de qualidade de dados. Podem ser definidas como restrições que validam dados e relacionamentos entre os dados e são desenvolvidas por meio de linguagens de programação.

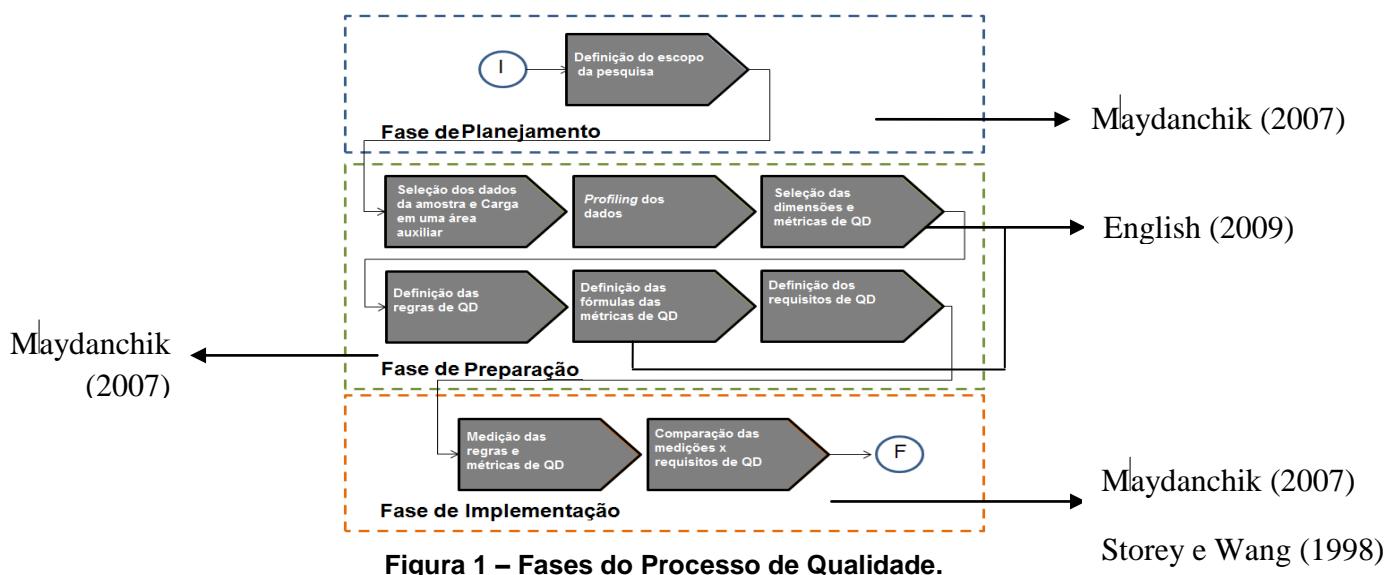
Fase de Refinamento das Regras de Qualidade: são validadas por especialistas em qualidade de dados e as regras de qualidade são aprimoradas para atingir o máximo da precisão com relação à identificação de erros. É uma fase que exige a participação de usuários do negócio.

Diferentemente das propostas anteriores que evidenciam a fase de definição das dimensões de QD a serem tratadas, Maydanchik (2007) destaca a preparação de um ambiente adequado à definição das regras de qualidade para a avaliação de dados.

A unificação de todas as propostas apresentadas, considerando partes de cada uma delas e compondo uma nova proposta, serviram de insumo para construir um processo de avaliação de dados que é a contribuição desse trabalho.

2. METODOLOGIA

A avaliação a ser feita neste estudo utilizou as três primeiras fases da proposta de Maydanchik (2007): fase de planejamento para definir o cenário em estudo, a fase de preparação para montar um ambiente de teste e, a fase de implementação para avaliar o estado dos dados na amostra de dados selecionada. A figura 1 resume o processo de qualidade da pesquisa, considerando cada etapa executada e as referências utilizadas para cada etapa.



O cenário em estudo é bem delimitado, pois é possível saber exatamente quais as tabelas de dados, registros e campos são relevantes e conforme citado por Maydanchik (2007) essa é uma das opções de escopo. Pode-se trabalhar com cenários mais amplos, porém não é o foco dessa pesquisa.

Para efetuar a fase de preparação, três atividades foram definidas:

Atividade 1: Carga dos dados de interesse em uma área auxiliar de trabalho;

Atividade 2: *Profiling* dos dados;

Atividade 3: Definição das dimensões, métricas e regras de qualidade.

As atividades 1 e 2 seguem a proposta de Maydanchik (2007).

A atividade 3 define dimensões e métrica de QD que não constam na proposta de Maydanchik (2007). Elas constam na proposta de English (2009).

Os dados da amostra possuem grandes volumes de informações, aproximadamente um milhão de registros mensais. Para executar a atividade 1, foram selecionados seis meses de cada tabela de dados dos sistemas em avaliação e efetuadas cópias dos dados em um diretório de uma estação de trabalho local da empresa. Os dados copiados foram armazenados em planilhas para posteriormente analisar e escrever as regras de qualidade. Somente os registros e campos de interesse foram copiados. Os meses mais recentes não foram utilizados, pois podem apresentar inconsistências de informações e dessa forma, podem ser reprocessados. Para não inviabilizar a pesquisa, gerando retrabalho em casos de dados que são reprocessados, foram escolhidos dados processados e validados há seis meses.

A seleção dos atributos foi feita juntamente com os usuários dos dados com base nas listas de atributos das estruturas físicas de cada sistema. Os atributos mais relevantes para o resultado do negócio estão descritos na tabela 1.

Tabela 1 – Atributos da Aplicação Risco Financeiro

| Sistema A | Sistema B |
|-----------------------------------|--|
| Código Produto | Código Operação |
| Número do Contrato | Código Produto |
| Código Segmento Mercado | Código Segmento mercado |
| Data Contratação Operação | Data Início Operação |
| Data Vencimento Operação | Data Vencimento Operação |
| Valor Líquido Operação Liquidação | Valor Saldo Devedor Contrato Cambio |
| Valor Contratado | |
| Sistema C | Sistema D |
| Código Operação | Código Identificação Modelo Estatístico LGDBest-Estimate |
| Código Produto | Código Identificação Modelo Estatístico PD |
| Código Segmento Mercado | Código Produto |
| Data Início Operação | Indicador Default |
| Data Vencimento Operação | Número Contrato Crédito |
| Valor Saldo Contábil Patrimônio | Número Contrato Título Derivativo |
| Código do Tipo de Registro | Valor EAD Econômica |
| | Valor LGD-Best-Estimate |
| | Valor PD PIT 12 Meses |

Na atividade 2, foram listadas as estruturas físicas de cada tabela de dados dos sistemas a serem avaliados, contendo os atributos, tipos, tamanho, precisão de casas decimais, chave primária, campos mandatórios, regras de preenchimento. Essa é a

atividade de *profiling*, conforme citada no processo de qualidade de Maydanchik (2007). As tabelas descritas no *profiling* não possuem um modelo de dados relacional. Elas estão modeladas de forma individual e não normalizadas.

Para executar a atividade 3, foram listados problemas de QD aparentes dos atributos e tabelas da amostra com base no *profiling* e análises mensais feitas pelos usuários da aplicação. Para cada problema relatado foram relacionadas dimensões de qualidade diretamente afetadas por esses problemas.

Na tabela 2 são listados os atributos e tabelas da aplicação, problemas e respectivas dimensões de qualidade afetadas.

Tabela 2 – Relação das tabelas e atributos da aplicação, problemas QD e dimensões de qualidade

| Tabela da Aplicação | Atributo da Aplicação | Problema de QD | | | | Dimensões de Qualidade | | | | | |
|---------------------|---|-------------------|--------------------|--------------------------|---------------------------|------------------------|----------------|----------------|----------|--------------|-----------|
| | | Dados incompletos | Domínios Inválidos | Duplicidade de registros | Atraso na Disponibilidade | Completude | Confiabilidade | Valor agregado | Acurácia | Consistência | Unicidade |
| B,C | Código da Operação | X | X | X | | X | X | X | X | X | |
| A,D | Número Contrato Crédito | | X | X | | | X | | X | X | X |
| A,D | Número Contrato Título Derivativo | | X | X | | | X | | X | X | X |
| A,B,C | Código Segmento Mercado | X | X | | | X | X | X | X | X | |
| A,B,C,D | Código Produto | | X | X | | | X | | X | X | X |
| A | Data Contratacao Operacao | X | X | | | X | X | X | X | X | |
| B,C | Data Início da Operação | X | X | | | X | X | X | X | X | |
| A,B,C | Data Vencimento da Operação | X | X | | | X | X | X | X | X | |
| D | Código Identificacao Modelo LGD-Best-Estimate | X | X | | | X | X | X | X | X | |
| D | Código Identificacao Modelo Probability-of-Defau | X | X | | | X | X | X | X | X | |
| D | Indicador Default | X | X | | | X | X | X | X | X | |
| A | Valor Liquido Operacao Liquidacao | X | X | | | X | X | X | X | X | |
| A | Valor Contratado | X | X | | | X | X | X | X | X | |
| B | Valor Saldo Devedor Contrato Cambio | X | X | | | X | X | X | X | X | |
| C | Valor Saldo Contabil Patrimonio | X | X | | | X | X | X | X | X | |
| D | Valor Exposition-at-Default Economica | X | | | | X | X | X | | | |
| D | Valor LGD-Best-Estimate | | X | X | | X | X | X | X | X | |
| D | Valor Probability-of-Default Point-in-Time 12 Mes | X | X | | | X | X | X | X | X | |
| C | Código da Operação | | | X | | | | | | | X |
| D | Código do Tipo de Registro | | | | X | | | | | | X |
| D | Todos os atributos | | | | | X | | | | | X |

Ao definir as dimensões de qualidade, é possível associar métricas de QD a cada dimensão. As dimensões de qualidade confiabilidade e valor agregado citadas na tabela 2 não tiveram suas métricas de QD associadas conforme tabela 3, pois não foram medidas nesta pesquisa. Para cada métrica foi definida uma fórmula a fim de efetuar as medições.

Tabela 3 – Métricas QD da amostra

| Métrica | Fórmula |
|-----------------|--|
| Acurácia | $\frac{\text{número de valores desejáveis (*)}}{\text{número total de registros da amostra}} \times 100$ <small>(*) Valores desejáveis corresponde a uma lista de valores válidos definidos de acordo com as regras de negócio.</small> |
| Consistência | $\frac{\text{número de valores corretos (**)}}{\text{número total de registros da amostra}} \times 100$ <small>(**) Valores corretos são consistências de valores ou regras específicos de acordo com as regras de negócio.</small> |
| Completude | $\frac{\text{número de valores preenchidos}}{\text{número total de registros da amostra}} \times 100$ |
| Unicidade | Número total de registros repetidos da chave primária. Chave primária = código do tipo de registro + código da operação |
| Disponibilidade | $\frac{\sum_{1,6} (\text{Tempo real de entrega dos dados} - \text{Tempo desejável de entrega dos dados})}{6}$ <small>Sendo 6 = quantidade de meses avaliados</small> |

Para esta pesquisa foram feitas duas abordagens de avaliação para escrever regras de qualidade com base na proposta de Maydanchik (2007).

A primeira abordagem avalia os atributos das tabelas. Essas regras validam atributos individuais e dois atributos que tenham dependência entre si. Os atributos foram inspecionados de acordo com a restrição de domínios de atributos, especificada para cada um deles. Atributos que se relacionam entre si foram avaliados com restrições de dependência entre eles.

A segunda abordagem avalia tabelas da amostra, nesse caso, os atributos são avaliados em conjunto, validando a tabela como um todo. Para avaliar tabelas da amostra, analisou-se a integridade existencial de acordo com os campos candidatos à uma chave primária.

As regras de qualidade utilizadas foram escritas na linguagem de consulta a banco de dados *SQL* (*structured query language*) e executadas dentro do banco de dados da aplicação de estudo.

Definidos os atributos da amostra, as regras e métricas de QD, é possível definir os requisitos de qualidade para cada atributo e para as tabelas avaliadas.

Os requisitos de QD definem os resultados esperados para cada métrica de QD de acordo com as expectativas dos usuários. Para a definição desses requisitos, foram consultados os usuários da aplicação. Eles foram denominados nesta pesquisa como valores *targets* para cada métrica de QD. Os valores *targets* das métricas QD definidos foram **100%** para grande parte dos atributos.

3. RESULTADOS

A fase de implementação da pesquisa iniciou com a execução de cada regra de qualidade definida para os atributos e tabelas da amostra. Os resultados obtidos foram armazenados em planilhas. As fórmulas das métricas de QD foram escritas na ferramenta Excel, pois os dados resultantes das regras de qualidade que serviram como

valores de entrada para as métricas já estavam armazenados em planilhas e essa ferramenta tornou-se mais prática para o cenário em questão. O resultado de cada métrica foi armazenado em planilhas também.

Para finalizar, os valores medidos em cada métrica e os valores *targets* foram comparados para verificação se os dados avaliados estão de acordo com os requisitos de QD. A comparação dos valores e os resultados finais foram efetuados na ferramenta Excel também. As tabelas 4 e 5 apresentam os resultados das medições.

Tabela 4 – Medição dos Atributos.

| | DIMENSÃO DE QUALIDADE | | | | | |
|-------------------------------------|-----------------------|--------|------------|--------|--------------|--------|
| | Acurácia | | Completure | | Consistência | |
| | Medida | Target | Medida | Target | Medida | Target |
| Sistema A | | | | | | |
| Código Produto | | | | | 100,0% | 100,0% |
| Número do Contrato | | | | | 100,0% | 100,0% |
| Código Segmento Mercado | 93,4% | 100,0% | 93,4% | 100,0% | | |
| Data Contratação Operação | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% | 100,0% |
| Data Vencimento Operação | | | 100,0% | 100,0% | 100,0% | 100,0% |
| Valor Líquido Operação Liquidacão | | | 100,0% | 100,0% | 100,0% | 96,0% |
| Valor Contratado | | | 100,0% | 100,0% | 100,0% | 100,0% |
| Sistema B | | | | | | |
| Código Operação | 83,4% | 100% | | | 100,0% | 100% |
| Código Produto | 6,08% | 100% | | | 99,90% | 100% |
| Código Segmento mercado | 86,3% | 100% | 86,3% | 100% | | |
| Data Início Operação | 99,9% | 100% | 100,0% | 100% | 99,99% | 100% |
| Data Vencimento Operação | | | 100,0% | 100% | 99,7% | 100% |
| Valor Saldo Devedor Contrato Cambio | | | 100,0% | 100% | 100,0% | 100% |

Tabela 5 – Medição das Tabelas.

| | DIMENSÃO DE QUALIDADE | | | |
|---|-----------------------|--------|-----------------|--------|
| | Unicidade | | Disponibilidade | |
| | Medida | Target | Medida | Target |
| Sistema C | | | | |
| Código Operação | 93,5% | 100% | | |
| Código Tipo de Registro | | 100% | | |
| Sistema D | | | | |
| Código Identificação Modelo LGD-Best-Estimate | | | | 100% |
| Código Identificação Modelo Probability-of-Default | | | | 100% |
| Código Produto | | | | 100% |
| Indicador Default | | | | 100% |
| Número Contrato Crédito | | | | 100% |
| Número Contrato Título Derivativo | | | | 100% |
| Valor Exposition-at-Default Econômica | | | | 100% |
| Valor LGD-Best-Estimate | | | | 100% |
| Valor Probability-of-Default Point-in-Time 12 Meses | | | | 100% |

A comparação entre os resultados das métricas (coluna Medida) e os requisitos de QD (coluna *Targets*) demonstram o estado atual dos dados.

Na tabela 4, a dimensão de qualidade que mais apresentou resultados diferentes entre os valores medidos e os valores esperados (*targets*) foi a acurácia. Isso demonstra que os valores da aplicação de estudo não são padronizados conforme o esperado e são preenchidos de acordo com regras não identificadas durante a análise dos dados. Cada sistema adota um critério de preenchimento. O conjunto de valores válidos foi definido juntamente com os usuários de dados, mas não refletiram todas as regras aplicadas nos dados atualmente.

Os resultados das métricas unicidade e disponibilidade que avaliaram os sistemas C e D respectivamente não se apresentaram de acordo com os requisitos de QD.

O sistema C necessita de uma integridade existencial no seu modelo de dados, pois apresenta registros com valores duplicados para os atributos avaliados e candidatos à chave primária. Na amostra avaliada, o número de valores não duplicados correspondeu a 93,5% do total de registros avaliados e o valor esperado pelos usuários (*target*) era 100% de unicidade conforme descrito na tabela 5. O tratamento dos registros duplicados antes de gravar os registros na tabela são manuais e as regras de negócio devem ser revistas para que seja possível aplicar uma regra automática de retirada de registros duplicados, enquanto os dados apresentarem repetições, ou seja, baixa qualidade.

O sistema D apresentou uma disponibilidade de acesso de seus dados abaixo do esperado ao atingir 41,7%, pois conforme consta na tabela 5 o valor esperado (*target*) era 100%. Os sistemas origens devem ser analisados para anteciparem o tempo de entrega dos seus dados para o sistema D, pois a baixa disponibilidade desses sistemas origens afeta o sistema D também.

4. CONCLUSÕES

O processo de qualidade aplicado na pesquisa se encerra na fase de implementação, porém outras fases podem dar continuidade, conforme a proposta de Maydanchik (2007), como a fase de refinamento das regras de qualidade e fase de avaliação contínua dos dados.

Maydanchik (2007) não menciona as dimensões de qualidade citadas na literatura para construir métricas. O autor visa escrever regras com maior precisão. Ele define em sua proposta duas métricas para medição dos dados: completude e a acurácia. Essas métricas são apresentadas somente na fase de implementação. Nesta pesquisa, essa ordem foi alterada, apresentando as métricas utilizadas na fase de preparação. Outra alteração efetuada foi a definição das regras de qualidade na fase de preparação também e não conforme a proposta de Maydanchik (2007) na fase de implementação.

A pesquisa não contemplou a avaliação do modelo de dados, pois atualmente, os dados da aplicação de risco financeiro estão definidos em tabelas não relacionadas do banco de dados. O modelo atual dos dados define a estrutura física das tabelas: atributos, tipos de dados, tamanho, obrigatoriedade dos campos e chaves primárias, visando dois requisitos básicos de QD, completude e unicidade.

Numa segunda fase da aplicação em estudo, o modelo de dados deverá sofrer uma reestruturação para relacionar e normalizar as tabelas, deixando de ser um simples repositório de dados. Devido a essas características, nesta pesquisa não foram definidas e avaliadas regras de QD para validação da integridade relacional de dados. A avaliação se concentrou na integridade individual dos atributos e tabelas relevantes, principalmente os atributos que as aplicações seguintes utilizam como variáveis para o cálculo de risco financeiro.

De um modo geral, o processo de avaliação de dados retratou como os dados se encontram momentaneamente e pode ser aplicado sempre que desejado, antes dos dados serem consumidos, substituindo a análise manual efetuada atualmente pelos usuários dos dados.

A vantagem na avaliação dos dados antes de serem disponibilizados é clara, pois detectados problemas de qualidade que inviabilizam a utilização dos dados, os sistemas origens devem ser acionados para a correção sem que os sistemas destinos tenham efetuados seus processamentos. O tempo de avaliação dos dados e a correção nos sistemas fontes é ainda menor que o tempo necessário para o processamento de todos os sistemas, análises visuais nos dados, correções pontuais e reprocessamento dos dados quando necessário. A principal contribuição deste trabalho foi a unificação de propostas pesquisadas, gerando um processo para avaliar os dados.

Como sugestões de trabalho futuros, podem ser citados a avaliação de cenários com modelos de dados relacionais, definindo regras de qualidade para integridade relacional; automatização do processo de avaliação dos dados; estudo e análise do custo em termos de tempo de desenvolvimento para aplicação de um processo de avaliação de QD nos projetos de dados.

5. REFERÊNCIA BIBLIOGRÁFICA

- ENGLISH, L. P. (2009) **Information Quality Applied - Best Practices for Improving Business Information, Processes, and Systems**, Indianapolis, Indiana, Wiley Publishing, Inc, pp. 57-245.
- MAYDANCHIK (2007), A., **Data Quality Assessment - Data quality for Practitioners**, Technics Publications, LLC, pp. 169-309.
- STOREY, V., WANG, R. Y. (1998) **Modeling Quality Requirements in Conceptual Database Design**, Proceedings of the 1998 Conference on Information Quality, October. pp. 64-87.
- WANG, R. Y., KON, H., MADNICK, S. (1993), **Data Quality Requirements Analysis and Modeling**, Proceedings of the Ninth International Conference of Data Engineering, April. pp. 670-677.

aper:152847_1

Utilizando Técnicas de *Data Science* para Definir o Perfil do Pesquisador Brasileiro da Área de Ciência da Computação

Gláucio R. Vivian¹, Cristiano R. Cervi¹

¹Instituto de Ciências Exatas e Geociências (ICEG)
Universidade de Passo Fundo (UPF) – Passo Fundo – RS – Brazil

{149293, cervi}@upf.br

Abstract. In this paper we collect the information in the Lattes Curriculum of 382 Brazilian researchers CNPq productivity to the area of Computer Science. This information was stored in a XML file. We propose a approach to identify the profile of researchers using Data Science tecniques. The largest contribution presented is an improvement in the Rep-Index metric. Such improvement allows better classification of researchers from the original index. We found some disparities in favor of males and more developed regions such as the Southeast and South of Brazil.

Resumo. Neste artigo coletamos as informações do Currículo Lattes de 382 pesquisadores brasileiros de produtividade do CNPq para a área de Ciência da Computação. Essas informações foram armazenadas em um arquivo XML. Propomos uma abordagem a fim de identificar o perfil dos pesquisadores utilizando técnicas de Data Science. A maior contribuição apresentada é uma melhoria na métrica Rep-Index. Tal melhoramento permite aumentar a qualidade da classificação dos pesquisadores em relação ao índice original. Encontramos algumas disparidades a favor do gênero masculino e regiões mais desenvolvidas como o Sudeste e Sul.

1. Introdução

Com a expansão de conteúdos disponibilizados na web, a quantidade dos dados não estruturados cresceu vertiginosamente. Os idealizadores da web inicialmente se preocuparam apenas com a apresentação dos dados. Mais recentemente, com o advento do conceito de web semântica, passou-se a inverter essa situação. Agora, além da apresentação, a estruturação dos dados disponíveis na web tornou-se relevante. Nesse contexto, surgiram as tecnologias para definirem um formato padronizado de troca de informações em arquivos semiestruturados(XML).

No Brasil os pesquisadores do Conselho Nacional de Desenvolvimento Científico e Tecnológico(CNPq)¹ devem possuir um currículo de acesso público na plataforma LATTES². Através dessa plataforma centralizada, as instituições e agências de fomento tem acesso às informações relativas a toda trajetória acadêmica do pesquisador. Tais informações são indispensáveis para a realização de estudos de bibliometria, cientometria, perfis de pesquisadores, métricas de avaliação e redes de colaboração. Essas informações

¹<http://www.cnpq.br>

²<http://lattes.cnp.br>

também são úteis para traçar estratégias a fim de promover o desenvolvimento científico e tecnológico do campo em estudo.

O objetivo deste trabalho é analisar os arquivos XML da plataforma Lattes dos pesquisadores de produtividade do CNPq para a área de Ciência da Computação. A análise possibilita a construção do perfil dos pesquisadores. Utilizamos técnicas de *Data Science* para analisar os dados. Agregado a análise dos dados, propomos um melhoramento na métrica Rep-Index[Cervi et al. 2013a] específica para a área da Ciência da Computação.

Este artigo está organizado da seguinte forma: Seção 2 são analisados alguns trabalhos relatados. Seção 3 é exposta a abordagem proposta. Na seção 4 são apresentados os experimentos e resultados encontrados. Finalmente, na seção 5, são apresentadas as conclusões e sugestões de trabalhos futuros.

2. Trabalhos Correlatos

Encontramos diversos estudos sobre a coleta de informações acadêmicas. No trabalho de [Mena-Chalco and Junior 2009] é apresentada uma ferramenta para recuperar dados acadêmicos diretamente da plataforma Lattes. Trata-se do *scriptLattes*, uma ferramenta que recupera os dados de páginas HTML. O SOSLattes proposto por [Galego 2013] em sua dissertação de mestrado é aprimoramento do *scriptLattes* com foco na utilização de ontologias. O trabalho de [Alves et al. 2011] propõem a ferramenta *LattesMiner*. O utilitário faz parte de um projeto maior chamado SUCUPIRA(Sistema Unificado de Currículos e Programas: Identificação de Redes Acadêmicas) que tem o aporte financeiro da CAPES. O seu objetivo é permitir a extração de dados com um alto nível de abstração. A principal diferença com relação a outros extratores está no fato de utilizar o nome do pesquisador como critério de busca e não a identificação única(ID).

Nos trabalhos de [Cervi et al. 2013a] [Cervi et al. 2013b] foi realizado um estudo com os perfis de pesquisadores de produtividade do CNPq de três áreas distintas: computação, odontologia e economia. Os dados foram coletados da plataforma Lattes, DBLP, Microsoft Academic Search, Arnetminer, Google Scholar e Publish or Perish. Tal estudo propõe uma nova métrica para comparar os perfis dos pesquisadores. A métrica Rep-Index identifica a reputação dos pesquisadores usando um conjunto de indicadores. Através de um número entre 1 e 5 busca-se a classificação da reputação de um pesquisador considerando 18 elementos da sua carreira acadêmica e científica. O cálculo da métrica é realizado através da média ponderada, dessa forma cada elemento do perfil do pesquisador possui um peso que pode ser ajustado de acordo com a realidade da área em questão. A abordagem proporciona uma análise mais equilibrada da trajetória acadêmica do pesquisador, pois envolve diversos elementos da carreira científica. Para maiores detalhes vide [Cervi et al. 2013a] [Cervi et al. 2013b].

No trabalho de [Arruda et al. 2009] é realizado um estudo sobre os pesquisadores da Ciência da Computação no Brasil. A motivação do estudo é descobrir a participação do público feminino e as distribuições geopolíticas. A metodologia consistiu em selecionar todos os programas de pós-graduação recomendados pela Capes, com conceito mínimo de 3. A partir disso coletaram-se todas as publicações presentes no Currículo Lattes entre 2000 e 2006. Com base em um conjunto predefinido de palavras-chaves se empregou uma análise estatística baseada no teste de Chi-quadrado de Pearson[Plackett 1983]. Fo-

ram apresentados os resultados e ao final se concluiu que o público feminino tem maior afinidade com algumas áreas de pesquisa, especialmente as que não apresentam componentes tecnológicos como redes e hardware. Também se observou que nessas áreas o público feminino é mais produtivo que o masculino. O trabalho foi limitado a estudar pesquisadores da computação que atuam em seus departamentos, contudo existem inúmeros pesquisadores que atuam em outros departamentos e áreas. Uma segunda limitação do estudo são as coautorias contadas em duplicidade que não foram retiradas nos totais gerais e podem modificar os resultados.

[Wainer et al. 2009] refizeram parte de uma análise anteriormente publicada no ano de 1995. Inicialmente o artigo apresenta a metodologia e resultados obtidos na publicação anterior. Neste novo estudo experimental avaliou-se de forma quantitativa 147 artigos científicos da Ciência da Computação, selecionados randomicamente nas publicações da ACM(*Association for Computing Machinery*)³ durante o ano 2005. Foram definidas as seguintes categorias para classificação: teórica, design e modelagem, estudos empíricos, testes de hipóteses e outros. No estudo anterior utilizaram-se quatro revisores de forma randômica para a classificação. Neste novo estudo elas foram realizadas aos pares para o mesmo artigo. Nos casos de discrepâncias entre os revisores foi iniciada uma discussão entre os mesmos para chegar em conjunto a uma classificação final. O estudo apresenta os resultados da classificação e conclui apontando que o percentual de pesquisas nas áreas experimentais e empíricas não avançou significativamente em relação ao estudo anterior.

3. Abordagem Proposta

Nesta seção definimos uma proposta de abordagem com o intuito de analisar os arquivos XML dos pesquisadores. Para isso definimos uma metodologia de trabalho, bem como a construção do *dataset* e as técnicas utilizadas na análise dos dados.

3.1. Metodologia

Definiu-se uma metodologia com o objetivo de realizar um experimento prático com dados reais obtidos através do *ScriptLattes*. Elencaram-se os critérios mais importantes com base na leitura dos trabalhos relatados sobre a definição de perfis acadêmicos. Assim, são elaboradas consultas personalizadas para análise dos seguintes critérios:

1. Nível e gênero: permite avaliar a distribuição das bolsas por gênero do pesquisador e nível da bolsa.
2. Formação acadêmica: importante para avaliar a formação acadêmica dos pesquisadores.
3. Estado da federação: permite avaliar a distribuição das bolsas por UF. Possibilita a elaboração do índice de pesquisador por habitante por UF e região.
4. Média de anos após título de doutor: permite avaliar o tempo mínimo, médio e máximo para chegar a determinado nível considerando a data de término do Doutorado.
5. Projeções de orientação e produção: permite realizar uma estimativa em relação à produção acadêmica e o número de orientações.

³<https://www.acm.org/>

6. Índice de produção por orientações: o índice representa a razão entre a produção e orientação.

Além da análise com técnicas de *Data Science*, buscamos aprimorar o Rep-Index para que sejam utilizados somente os elementos mais relevantes para o conjunto de dados de cada área. Este melhoramento torna o Rep-Index mais abrangente e com critérios mais específicos. Inicialmente apresentamos uma previa análise dos dados para os critérios presentes no Rep-Index. Esta etapa tem por objetivo encontrar um subconjunto de critérios que são mais relevantes para a métrica Rep-Index. Ao final seremos capazes de tornar o seu cálculo específico para cada área por meio da aplicação de pesos identificados para cada critério.

3.2. Dataset

Inicialmente obteve-se a lista dos pesquisadores com bolsa de produtividade ativa para a área de Ciência da Computação no site do CNPq⁴. Nesta lista encontram-se algumas informações importantes tais como: nome, nível, instituição e data de início e término da bolsa de pesquisa. Foram encontrados 382 pesquisadores ativos com bolsa na data de 28/07/2015. Os pesquisadores contemplados com mais de uma bolsa foram considerados como sendo um único pesquisador.

Existem duas formas de obter as informações da plataforma Lattes. A primeira consiste em estabelecer um acordo institucional com o CNPq e obter os arquivos XML definidos com a ontologia da Comunidade Conscientias⁵. Outra possibilidade consiste na utilização de ferramentas de recuperação de informações através da *web*. Conforme visto na seção 2, encontramos diversos trabalhos com esse objetivo. O mais consistente é o trabalho de pesquisadores da USP [Mena-Chalco and Junior 2009] chamado de *scriptLattes*⁶. Tal ferramenta realiza a recuperação das informações do currículo utilizando os arquivos HTML da página pessoal de cada pesquisador na plataforma. Os resultados finais são vários arquivos, gráficos, mapas de geolocalização, grafos de cooperação e tabelas. Também se encontra um arquivo XML definido com uma ontologia muito semelhante ao disponibilizado pela plataforma Lattes. O trabalho realizado pelo *scriptLattes* não depende de acordos institucionais. Dessa forma optou-se pelo mesmo, pois encontramos diversas publicações [Mena-Chalco and Junior 2009] [Mena-Chalco et al. 2012] que indicam a sua qualidade e credibilidade.

A partir da lista com os nomes dos pesquisadores da plataforma Lattes, utilizou-se a API de pesquisa do Google[Netti 2010] tendo como critério de pesquisa o nome do pesquisador e uma restrição para limitar as buscas ao domínio lattes.cnpq.br. A página da plataforma Lattes não permite o acesso a indexadores de conteúdo. Mesmo assim a maioria dos pesquisadores possuem páginas pessoais/institucionais que apresentam links referenciando os seus currículos. Dessa forma o buscador Google obteve a ID de aproximadamente 85% dos pesquisadores. O restante foi coletado de forma manual através de pesquisa no *site* do Lattes. De posse das identificações únicas de cada pesquisador, utilizou-se a ferramenta *scriptLattes* para obter as informações da plataforma em formato semiestruturado(XML). As publicações científicas foram limitadas ao período de 2004

⁴http://plsql1.cnpq.br/divulg/RESULTADO_PQ_102003.curso

⁵<http://lmp1.cnpq.br/lmp1/>

⁶<http://scriptlattes.sourceforge.net>

e 2014. Os dados obtidos não foram validados pois se presume que a responsabilidade sobre os mesmos é do pesquisador. Erros e inconsistências podem ocorrer, pois há um processo limitado de validação dos dados na plataforma Lattes. A fim de confirmar que os dados na plataforma Lattes encontram-se atualizados, analisou-se os mesmos com base na data da última atualização de cada pesquisador. Observou-se que aproximadamente 93% dos pesquisadores atualizaram as informações no ano de 2015.

Para avaliarmos os perfis dos pesquisadores foi construído um *dataset* com base nas informações recuperadas. A maioria das informações encontradas apresentam ligeiras diferenças ortográficas devidas principalmente a abreviações ou diferentes tabelas de caracteres. A desambiguação dos nomes próprios e títulos foi feita utilizando o algoritmo da Distância de *Levenshtein* sem pesos⁷ e com uma prévia classificação considerando a primeira letra contida no seu título. Esta técnica foi proposta por [Mena-Chalco et al. 2012] com o objetivo de corrigir inconsistências encontradas na recuperação de informação. Consideram-se equivalentes os itens que apresentaram pelo menos 85% de similaridade na comparação de strings.

3.3. Filtragem, Limpeza e Análise dos dados

A partir do *dataset* se passou para etapa de filtragem e limpeza dos dados no arquivo XML. Foi escolhida a tecnologia XQuery em conjunto com o *software* BaseX⁸ para esta tarefa. A sua semântica e poder de expressão são equiparáveis à linguagem SQL para banco de dados relacionais. Nos trabalhos de [de Campos et al. 2014] [Kilpeläinen 2012] podemos comprovar a eficiência da linguagem na recuperação de informações em arquivos XML. No trabalho de [Silva et al. 2008] encontramos um estudo abrangente sobre técnicas de comparações de similaridade de *strings* usando a linguagem XQuery. Para a análise estatística dos dados se utilizou a linguagem R em conjunto com a IDE R-Studio⁹. O aprendizado de máquina foi realizado utilizando o *software* Weka¹⁰. Os resultados obtidos das análises serão relatados a seguir.

4. Experimentos e Resultados

Nesta seção são detalhados os experimentos e os resultados obtidos para abordagem proposta na seção 3.

4.1. Nível e Gênero

Observou-se um total de 75,92%(290) pesquisadores do CNPq para o gênero masculino. Apenas 24,08%(92) pesquisadores de gênero feminino. Isto indica se tratar de uma área de predominância masculina. O CNPq atribui os níveis¹¹ 1A, 1B, 1C, 1D e 2 para os pesquisadores. Também analisamos os totais por nível e gênero de forma conjunta. Na tabela 1 pode-se visualizar mais este critério de análise do perfil. Observa-se uma clara predominância do par: nível 2/gênero masculino em todos os níveis existentes.

⁷Explicação: Nesta situação, consideram-se os pesos iguais a 1 para inserção, remoção e alteração.

⁸<http://basex.org>

⁹<http://www.rstudio.com/>

¹⁰<http://www.cs.waikato.ac.nz/ml/weka/>

¹¹Critérios: http://cnpq.br/web/guest/view/-/journal_content/56_INSTANCE_0oED/10157/49290

Tabela 1. Distribuição dos pesquisadores por nível e gênero

| Nível | SR | 2 | | 1D | | 1C | | 1B | | 1A | |
|--------|-------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| Quant. | 1 | 244 | | 52 | | 37 | | 24 | | 24 | |
| Perc. | 0,26% | 63,87% | | 13,61% | | 9,69% | | 6,28% | | 6,28% | |
| Gênero | M | M | F | M | F | M | F | M | F | M | F |
| Quant. | 1 | 183 | 61 | 37 | 15 | 29 | 8 | 20 | 4 | 20 | 4 |
| Perc. | 0,26% | 47,91% | 15,97% | 9,69% | 3,93% | 7,59% | 2,09% | 5,24% | 1,05% | 5,24% | 1,05% |

4.2. Curso de Formação Acadêmica

Analisamos a formação acadêmica dos pesquisadores a fim de melhor determinar o perfil dos pesquisadores. Agrupamos as áreas de formação em: Computação(Ciência da Computação, Informática, Análise de Sistemas e Sistemas de Informação), Engenharias(Elétrica, Eletrônica, Civil, Nuclear, Produção, Mecânica e Computação), Matemática(Licenciatura e Bacharelado), Física(Aplicada e Computacional) e Outras áreas. Nos cursos realizados no exterior procurou-se utilizar a nomenclatura equivalente no Brasil. Na tabela 2 pode-se visualizar a distribuição dos pesquisadores de acordo com a sua formação acadêmica.

Tabela 2. Distribuição dos pesquisadores por Formação Acadêmica

| Curso | Graduação | Especialização | Mestrado | Doutorado |
|--------------|-------------|----------------|-------------|-------------|
| Computação | 188(47,47%) | 27(62,79%) | 253(64,05%) | 234(60,15%) |
| Engenharias | 131(33,08%) | 4(9,30%) | 79(20,00%) | 101(25,96%) |
| Matemática | 53(13,38%) | 2(4,65%) | 43(10,89%) | 18(4,63%) |
| Física | 10(2,53%) | | 11(2,78%) | 11(2,83%) |
| Outras áreas | 14(3,54%) | 10(23,26%) | 9(2,28%) | 25(6,43%) |
| Totais | 396(100%) | 43(100%) | 395(100%) | 389(100%) |

Também fizemos uma análise da formação utilizando algoritmos de associação. O algoritmo Apriori[Agrawal et al. 1994] foi o que apresentou melhores resultados configurado com parâmetro confiança maior que 85%. Este algoritmo busca encontrar associações do tipo causa ==> consequência. Na figura 1 pode-se visualizar as regras de associação encontradas.

1. graduacol=computacao mestradol=computacao 160 ==> doutoradol=computacao 147 conf: (0.92)
2. graduacol=computacao doutoradol=computacao 170 ==> mestradol=computacao 147 conf: (0.86)
3. graduacol=computacao 198 ==> doutoradol=computacao 170 conf: (0.86)

Figura 1. Associações encontradas na Formação Acadêmica

Tanto na distribuição dos pesquisadores quanto na busca por regras de associação, observa-se a predominância da formação na área de Ciência da Computação. Com relação às associações, constata-se que quando a causa é graduação em computação, existe uma forte consequência da formação de doutorado ser realizada na mesma área.

4.3. Estado da Federação

A partir da distribuição de bolsas por estado da federação calculamos o índice de pesquisadores por habitante¹². Na figura 2 pode-se visualizar de forma crescente o índice em questão.

¹²Fonte dados populacionais: <http://www.ibge.gov.br/apps/populacao/projecao/>

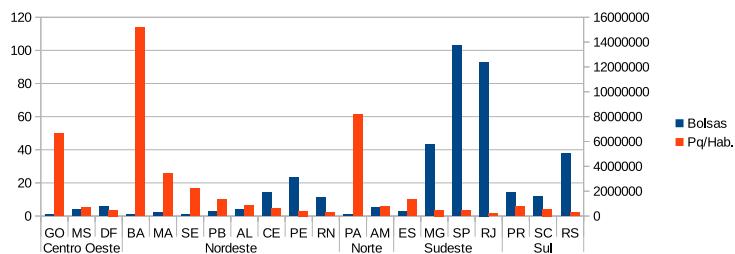


Figura 2. gráfico do índice de pesquisadores por habitante

Verifica-se que nos estados RJ, RS, RN, PE e SP encontram-se os melhores índices(valores menores) de pesquisadores por habitante. Por outro lado, os estados BA, PA, GO, SE e PB apresentam os piores índices(valores maiores) em relação ao restante das áreas geopolíticas da federação.

4.4. Média de Anos Após a Conclusão do Doutorado

Considera-se que a formação de um doutor seja um importante instrumento para a qualificação da pesquisa, para o desenvolvimento de novos cientistas, bem como da possibilidade deste doutor disseminar seu conhecimento por meio de seus novos projetos e captação de novos estudantes que queiram seguir a carreira científica. Na figura 3 pode-se visualizar o gráfico que demonstra o tempo médio após o doutorado em cada nível de bolsa. Removeu-se o nível SR pois o mesmo possui apenas um pesquisador, portanto a média não pode ser calculada. Os dados entre parênteses representam o total de bolsas em cada nível. Na figura 4 pode-se visualizar o histograma de anos após a conclusão do doutorado.

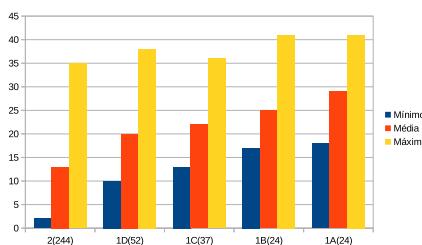


Figura 3. Mínimo, média e máximo de tempo após término Dr

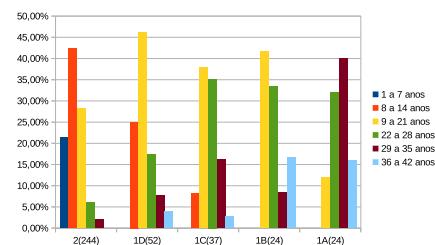


Figura 4. Histograma de tempo após término Dr

Observa-se que conforme o nível cresce, a média de anos também acompanha o seu crescimento. Com relação ao histograma, podemos observar a predominância de um tempo maior após o término do doutorado nos níveis superiores(1A e 1B).

4.5. Projeções de Orientação e Produção

Agruparam-se os quantitativos em: produção acadêmica(artigos, livros, capítulos de livros, produção bibliográfica, trabalhos em congressos, trabalhos técnicos e resumos expandidos) e orientações(TCC, mestrado, doutorado e iniciação científica). Após aplicou-se a técnica de regressão linear do Weka para projetar o próximo triênio(2015 a 2017). A equação obtida para a orientação é: $Orientacao = 85,5091(Ano) - 170432,3089$ e para produção é: $Producao = 77,2091(Ano) - 151453,8817$. Na figura 5 pode-se visualizar a projeção para a orientação e na figura 6 pode-se visualizar a projeção para a produção. Os totais utilizam a escala da esquerda.

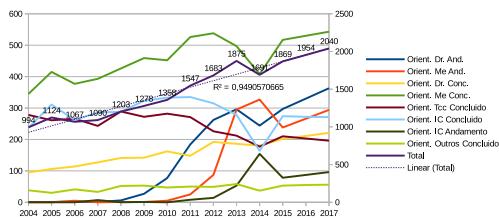


Figura 5. Projeção de orientações

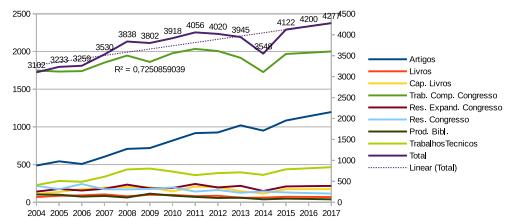


Figura 6. Projeção de produção

A partir da regressão linear observa-se que tanto a orientação quanto a produção estão em crescimento. Observando o coeficiente angular das equações, constamos que a orientação cresce ligeiramente superior a produção.

4.6. Índice de Produção por Orientações

Elaborou-se um quantitativo de produção e orientações por ano entre o período de 2004 e 2014. Após utilizou-se a técnica de regressão linear para realizar a projeção para o próximo triênio(2015 a 2017). A partir desses valores calculamos o índice de produção por orientação. Na figura 7 pode-se visualizar o índice mencionado.

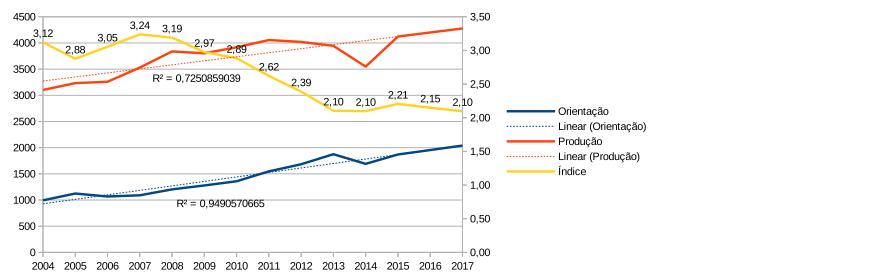


Figura 7. Índice de Produção por Orientações

Observa-se que o maior valor para o índice foi no ano de 2007(3,24). O gráfico mostra que este índice se encontra em queda e deve chegar à casa de 2,10 em 2017. Este fato pode ser interpretado como um ligeiro aumento na oferta de orientações, tal queda justifica-se pôr a orientação crescer ligeiramente mais do que a produção.

4.7. Rep-Index para a Computação

Com o objetivo de ajustar o Rep-Index para a área da Ciência da Computação, propomos um melhoramento com o intuito de atualizar os pesos originais da métrica mencionada. Inicialmente aplicamos os algoritmos ReliefF, GainRatio e ChiSquared para seleção de atributos. Para maiores detalhes sobre os algoritmos vide [Hall et al. 2009]. Os algoritmos resultam em um ranking, os valores são estao ajustados para média ponderada. Na tabela 3 pode-se visualizar os elementos do Rep-Index, pesos originais, *ranks* e os pesos propostos para cada técnica algorítmica mencionada.

A fim de mensurarmos a eficiência da proposta, calculamos a correlação de Spearman entre o nível do CNPq e o Rep-Index com os pesos ajustados. A aplicação desta correlação na comparação de rankings foi proposta por [Kozak and Bornmann 2012]. Para o Rep-Index original, obtivemos o valor de 0,3932. No caso do algoritmo ReliefF encontramos o valor de 0,4631. O GainRatio obteve 0,5298. Finalmente o ChiSquare

Tabela 3. Pesos para o Rep-Index

| Elemento | Rep-Index | | ReliefF | | GainRatio | | ChiSquared | |
|--|-----------|---------|---------|--------|-----------|----------|------------|--|
| | Original | Rank | Peso | Rank | Peso | Rank | Peso | |
| Grau de Instrução | 15 | 0,05733 | 10,63 | 0 | | | 0 | |
| Orientação de Mestrado | 4 | 0,04269 | 7,92 | 0,1167 | 6,25 | 95,0317 | 9,78 | |
| Orientação de Doutorado | 5 | 0,05597 | 10,38 | 0,2615 | 14 | 122,3346 | 12,59 | |
| Orientação de Pós-doutorado | 6 | 0,0177 | 3,28 | 0,0968 | 5,18 | 41,9503 | 4,32 | |
| Participação em Banca de Doutorado | 6 | 0,02988 | 5,54 | 0,1102 | 5,9 | 86,052 | 8,85 | |
| Participação em Banca de Mestrado | 4 | 0,03438 | 6,38 | 0 | | | 0 | |
| Membro de Corpo Editorial de Periódico | 5 | 0,02212 | 4,1 | 0,3664 | 19,62 | 74,191 | 7,63 | |
| Revisão de Periódico | 3 | 0,01991 | 3,69 | 0 | | | 0 | |
| Coordenação de Comitê de Conferência | 1 | 0,0122 | 2,26 | 0 | | | 0 | |
| Membro de Comitê de Conferência | 1 | 0,02292 | 4,25 | 0 | | | 0 | |
| Artigo em Periódico | 15 | 0,04147 | 7,69 | 0,2372 | 12,7 | 217,0859 | 22,33 | |
| Livro | 8 | 0,01122 | 2,08 | 0,0651 | 3,49 | 34,631 | 3,56 | |
| Capítulo de Livro | 5 | 0,01727 | 3,2 | 0,0732 | 3,92 | 38,3944 | 3,95 | |
| Trabalho Completo em Conferência | 8 | 0,02827 | 5,24 | 0,2835 | 15,18 | 99,3299 | 10,22 | |
| h-index | 7 | 0,05884 | 10,91 | 0,1412 | 7,56 | 75,6397 | 7,78 | |
| Rede de Coautoria | 3 | 0,03571 | 6,62 | 0,116 | 6,21 | 87,3619 | 8,99 | |
| Projeto de Pesquisa | 2 | 0,02434 | 4,51 | 0 | | | 0 | |
| Software | 2 | 0,00706 | 1,31 | 0 | | | 0 | |
| | 100 | 0,53928 | 100 | 1,8678 | 100 | 972,0024 | 100 | |

obteve o valor de 0,5339. Dessa forma, os pesos para o Rep-Index calculados com o algoritmo ChiSquare são os que possibilitaram os melhores resultados. A análise dos valores dos pesos para a área da Ciência da Computação nos possibilita verificar quais elementos do Rep-Index são mais relevantes no perfil dos pesquisadores da área.

5. Conclusões e Trabalhos Futuros

Os dados obtidos possibilitam identificar uma clara predominância do gênero masculino(76%) em todos os níveis. Com relação à formação acadêmica, a maioria apresenta formação na área de Computação, em segundo lugar as Engenharias, em terceiro a Matemática e em quarto lugar a Física. Com relação ao tempo de atuação como pesquisador, observou-se que em todos os níveis a média é superior ao tempo de produção científica regular exigida pela CAPES. Um fato interessante foi constatado no nível de acesso(nível 2), encontramos pesquisadores com apenas dois anos de conclusão do Doutorado. Tal fato indica que as regras estão permitindo a inserção de pesquisadores recém-formados e que apresentem uma produção científica regular produzida durante a sua formação acadêmica. Com relação ao índice de produção por orientação, projetado com a regressão linear para o próximo triênio, observamos que o mesmo tende a se aproximar de 2,10 em 2017. Tal fato justifica-se em função do aumento da projeção de orientação ser ligeiramente maior que o da produção.

A proposta de melhoria no Rep-Index se mostrou eficiente, principalmente para a classificação do nível 2 dos pesquisadores do CNPq. Pretendemos realizar um estudo mais abrangente incluindo outras áreas, algoritmos, correlações e dados mais atuais para comprovarmos a eficiência do melhoramento proposto. As informações obtidas sobre o perfil são o primeiro passo de um trabalho maior. Pretende-se no futuro propor um sistema de recomendação de publicações científicas com base nos perfis dos pesquisadores analisados e nos dados coletados da plataforma Lattes. Para tanto, iniciaremos um estudo sobre estratégias de recomendação. Neste contexto a construção do *dataset* e análise do perfil foram fundamentais para definirmos o escopo inicial dos próximos trabalhos.

Referências

- Agrawal, R., Srikant, R., et al. (1994). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499.

- Alves, A. D., Yanasse, H. H., and Soma, N. Y. (2011). Lattesminer: A multilingual dsl for information extraction from lattes platform. In *Proceedings of the Compilation of the Co-located Workshops on DSM'11, TMC'11, AGERE! 2011, AOOPES'11, NEAT'11, & VMIL'11, SPLASH '11 Workshops*, pages 85–92. ACM.
- Arruda, D., Bezerra, F., Neris, V., Rocha De Toro, P., and Wainera, J. (2009). Brazilian computer science research: Gender and regional distributions. *Scientometrics*, 79(3):651–665.
- Cervi, C. R., Galante, R., and Oliveira, J. P. M. d. (2013a). Application of scientific metrics to evaluate academic reputation in different research areas. *in: XXXIV International Conference on Computational Science(ICCS) 2013*. Bali, Indonesia.
- Cervi, C. R., Galante, R., and Oliveira, J. P. M. d. (2013b). Comparing the reputation of researchers using a profile model and scientific metrics. *in: XIII IEEE International Conference on Computer and Information Technology(CIT)*. Sydney, Australia.
- de Campos, L. M., Fernández-Luna, J. M., Huete, J. F., and Vicente-Lopez, E. (2014). Using personalization to improve xml retrieval. *Knowledge and Data Engineering, IEEE Transactions on*, 26(5):1280–1292.
- Galego, E. F. (2013). Extração e consulta de informações do currículo lattes baseada em ontologias. Master's thesis, Universidade de São Paulo, São Paulo - SP.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.
- Kilpeläinen, P. (2012). Using xquery for problem solving. *Software: Practice and Experience*, 42(12):1433–1465.
- Kozak, M. and Bornmann, L. (2012). A new family of cumulative indexes for measuring scientific performance. *PloS one*, 7(10):e47679.
- Mena-Chalco, J. P., Digiampietri, L. A., and Cesar-Jr, R. M. (2012). Caracterizando as redes de coautoria de currículos lattes. *Brazilian Workshop on Social Network Analysis and Mining (BraSNAM)*, pages 1–12.
- Mena-Chalco, J. P. and Junior, R. M. C. (2009). scriptlattes: an open-source knowledge extraction system from the lattes platform. *Journal of the Brazilian Computer Society*, pages 31–39. Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo – USP.
- Netti, K. (2010). Interactive guided online/off-line search using google api and json. *JCSI International Journal of Computer Science*, 7(5):167–174.
- Plackett, R. L. (1983). Karl pearson and the chi-squared test. *International Statistical Review/Revue Internationale de Statistique*, pages 59–72.
- Silva, M. E. V. d., Borges, E. N., and Galante, R. (2008). Xsimilarity: Uma ferramenta para consultas por similaridade embutidas na linguagem xquery.
- Wainer, J., Barsottini, C. G. N., Lacerda, D., and de Marco, L. R. M. (2009). Empirical evaluation in computer science research published by acm. *Information and Software Technology*, 51(6):1081–1085.

aper:152968_1

Workflows para a Experimentação em Análise de Similaridade de Imagens Médicas em um Ambiente Distribuído

Luis Fernando Milano-Oliveira¹, Matheus Peviani Vellone¹ e Daniel S. Kaster¹

¹Departamento de Computação – Universidade Estadual de Londrina (UEL)
Caixa Postal 10.011 – CEP 86057-970 – Londrina – PR – Brasil

{luismilano oliveira, matheusvellone}@gmail.com, dskaster@uel.br

Abstract. *Much of recent research is done towards improving Content-Based Medical Image Retrieval (CBMIR) systems. Among the open challenges, it is necessary to provide an interface with resources accessible for end users (like medical specialists) to define tasks related to image similarity. Furthermore, the high cost of image processing tasks as well as the always increasing sizes of the datasets demand that new solutions be scalable. Based upon these requirements, this work presents a proposal that makes use of a workflow system and a parallel processing framework to provide an environment in which end users can easily conduct experiments regarding image similarity over large datasets. A prototype of the environment was developed, allowing the definition of pipelines for image retrieval that are executed over Apache Spark, having achieved linear horizontal scalability in a tested sequence of tasks.*

Resumo. *Recentemente, tem-se realizado muitos esforços de pesquisa para a melhoria de Sistemas para a Recuperação de Imagens Médicas com Base em Conteúdo (CBMIR). Dentre os desafios em aberto, é necessário prover uma interface com recursos acessíveis aos usuários finais (como especialistas médicos) definirem tarefas relacionadas à similaridade de imagens. Além disso, o alto custo das tarefas relacionadas ao processamento de imagens, assim como a existência de conjuntos de dados cada vez maiores exigem que novas soluções sejam escaláveis. Com base nesses requisitos, este trabalho apresenta uma proposta que faz uso de um sistema de workflows e de um framework para processamento paralelo a fim de prover um ambiente em que usuários finais possam conduzir experimentos em similaridade de imagens sobre grandes conjuntos de dados. Foi desenvolvido um protótipo do ambiente, que permite definir pipelines de recuperação de imagens que são executados sobre o Apache Spark, tendo alcançado escalabilidade horizontal linear em uma sequência de tarefas avaliada.*

1. Introdução

A recuperação de imagens médicas por conteúdo (*Content-Based Medical Image Retrieval* – CBMIR) tem sido foco de um grande número de pesquisas nos últimos anos [Burak Akgül et al. 2011]. São diversos os desafios ainda em aberto para que as soluções existentes atinjam as expectativas de seus usuários.

Encontra-se na literatura de reconhecimento de padrões em imagens um número expressivo de trabalhos que descrevem *pipelines* de processamento a fim de obter resultados relevantes para domínios específicos de imagens [Wolf 2010]. Isto se explica

pela imensa quantidade de possibilidades a serem combinadas, incluindo algoritmos de extração de características, métodos de combinação e seleção/transformação de características e funções de distância, sendo que cada um desses elementos possui conjuntos de parâmetros e ponderações particulares. O objetivo principal desses trabalhos é reduzir a lacuna semântica existente entre as características reconhecidas automaticamente por um sistema e a percepção visual do usuário especialista.

Trabalhos como o de [Deserno et al. 2009] elencam algumas categorias de lacunas além da semântica que sistemas que busquem auxiliar a solução de problemas relacionados a CBMIR devem possuir, entre elas: lacunas de conteúdo, de características, de performance e de usabilidade. As lacunas de conteúdo envolvem o entendimento que o usuário possui de uma imagem, o que está diretamente ligado ao uso clínico que é dado ao sistema. As lacunas de características estão ligadas às limitações existentes nos métodos que são utilizados para representar as características de imagens numericamente. As lacunas de performance dizem respeito à velocidade de resposta de consultas feitas ao conjunto de dados, bem como à integração de um sistema com outros sistemas de informação utilizados no contexto médico. Em último lugar, as lacunas de usabilidade incluem problemas que usuários enfrentam ao utilizar os sistemas, bem como a dificuldade de customizar o processo de acordo com suas necessidades.

Neste trabalho é apresentada uma proposta de arquitetura baseada em *workflows* científicos e *frameworks* de processamento distribuído para a recuperação de imagens que leva em conta essas lacunas e busca fornecer ao usuário um ambiente onde ele possa realizar experimentos relacionados à similaridade de imagens médicas. Desta forma, o usuário tem acesso a recursos sofisticados de processamento de grandes volumes de dados por meio de uma interface intuitiva, que lhe permite combinar recursos de acordo com seu interesse e de forma escalável.

O restante do artigo está organizado conforme segue. A Seção 2 apresenta trabalhos correlatos. A Seção 3 apresenta a proposta deste trabalho, tanto no nível conceitual quanto de implementação. Na Seção 4 são descritos experimentos que foram realizados a fim de validar alguns aspectos da proposta. Por fim, a Seção 5 traz as conclusões e propostas de trabalhos futuros.

2. Trabalhos relacionados

Pode-se classificar os trabalhos relacionados em três grandes categorias: CBMIRs baseados em Sistemas de Gerenciamento de Banco de Dados (SGBDs), CBMIRs baseados em pipelines de operações e arquiteturas distribuídas para CBMIR.

Um exemplo de CBMIR baseado em SGBD é apresentado em [Bedo et al. 2012], que agrupa diversas técnicas já consolidadas de CBMIR e oferece uma ferramenta através da qual é possível realizar recuperação de imagens com base na percepção do usuário. A proposta dos autores é composta de diversos módulos, entre eles um módulo para a extração de características contendo vários extratores e um módulo para a interação do sistema proposto com um SGBD, que permite que sejam feitas consultas a um banco de dados contendo imagens médicas de um hospital. Essas consultas podem ser as tradicionais consultas que os SGBD suportam nativamente, consultas a metadados do formato DICOM (*Digital Imaging and Communications in Medicine*), além de consultas baseadas em conteúdo. O sistema Higiia também possui uma interface de usuário através da qual

foram feitos testes de classificação de mamografias.

Um trabalho recente na segunda categoria é o *framework* proposto em [Sridharan 2015], que permite a construção de *pipelines* para a análise de grandes coleções de imagens médicas. O autor apresenta como suas motivações o aumento de tamanho dos conjuntos de dados e também problemas criados pela baixa qualidade de imagens capturadas em ambiente clínico. A proposta do autor traz uma ferramenta que provê refinamento iterativo dentro do próprio *workflow*, o que possibilita o desenvolvimento de novos *pipelines* a partir de experimentação (combinação diferente de parâmetros). A interface de construção de *pipelines* se dá através de *scripts* na linguagem Python, a partir dos quais são geradas visualizações dos *workflows* em formato de grafo, que permitem que sejam feitas consultas a resultados intermediários e monitoramento sobre qual é o estado de execução de determinada tarefa.

Trabalhos na linha de arquiteturas distribuídas para CBMIR essencialmente empregam soluções já consolidadas para o processamento de grandes conjuntos de dados de forma paralela. Podem ser citados os trabalhos de [Jai-Andaloussi et al. 2013] e [Grace et al. 2014], onde os autores investigam a utilização do *framework* Hadoop como solução para se enfrentar problemas relacionados ao armazenamento e processamento de imagens médicas dentro de hospitais. Os resultados obtidos nesses trabalho apontam, além de uma melhora na performance, também a preservação da confidencialidade de pacientes e grande tolerância a falhas, através do suporte à redundância que o Hadoop traz.

O diferencial do presente trabalho é integrar vantagens das três categorias de trabalhos em uma única solução. Desta forma, usuários finais podem utilizar recursos de processamento distribuído de forma simples (como nos trabalhos da terceira categoria), por meio de *workflows* (que possuem flexibilidade e robustez na definição e execução de *pipelines*, como no trabalho da segunda categoria), e sem a necessidade de conhecimento aprofundado de programação (como o trabalho da primeira categoria).

3. Proposta de CBMIR baseada em Workflows e Execução Distribuída

A proposta deste trabalho consiste em possibilitar ao usuário construir o processo de definição do espaço de similaridade adequado a cada situação específica por meio da utilização de *workflows* na definição de *pipelines* para a recuperação de imagens médicas por conteúdo, tudo isso de maneira escalável. As subseções a seguir apresentam a arquitetura conceitual da proposta e aspectos de implementação.

3.1. Arquitetura Conceitual

A concepção da proposta consiste em aliar um ambiente para a definição de *workflows* cujas tarefas sejam relacionadas à recuperação de imagens por conteúdo e um ambiente para processamento e armazenamento distribuído que permita manipular eficientemente conjuntos de dados volumosos. Algumas características específicas importantes incluem:

- facilidade de interação com imagens no formato DICOM (o padrão mais utilizado no domínio médico [Larobina and Murino 2014]);
- o suporte para operações de manipulação das imagens, como a aplicação de filtros de realce, redução de ruídos, etc;

- a inclusão de um conjunto inicial de extratores de características que seja adequado para imagens médicas;
- a inclusão de um conjunto de seletores de características que possibilitem a redução de dimensionalidade dos vetores extraídos;
- a disponibilização de formas de análise de resultados para avaliar a adequação do *pipeline* gerado para o problema em questão.

Com base nestes elementos, a arquitetura conceitual proposta é apresentada na Figura 1. Na figura, o cliente define um *workflow* que é executado pelo motor de execução de *workflows*, que pode tanto estar integrado na interface de definição de *workflows* do cliente quanto estar alojado em um servidor dedicado a execução de *workflows*. O motor de execução realiza, então, chamadas a *web services* de acordo com as tarefas necessárias e estas, por sua vez, invocam as funcionalidades das bibliotecas de domínio específico para que sejam executadas de maneira distribuída em um *cluster*. Após o término do processamento, o resultado obtido é retornado até o cliente fazendo o mesmo caminho, em uma espécie de pilha. Os dados em si (imagens) são enviados para o *cluster* antes do início do processamento. Após o processamento de alguma tarefa, os resultados ficam armazenados no *cluster* para ser usados como entrada de outras tarefas. Embora o usuário possa transferir dados do *cluster* para o cliente, o caso de uso geral da proposta é que o cliente deve ser leve e o tráfego de dados entre cliente e servidor deve ser minimizado. Desta forma, o usuário pode definir a configuração do “experimento” e analisar a saída de forma iterativa, explorando variações de algoritmos e parâmetros utilizando o poder de processamento do *cluster*.

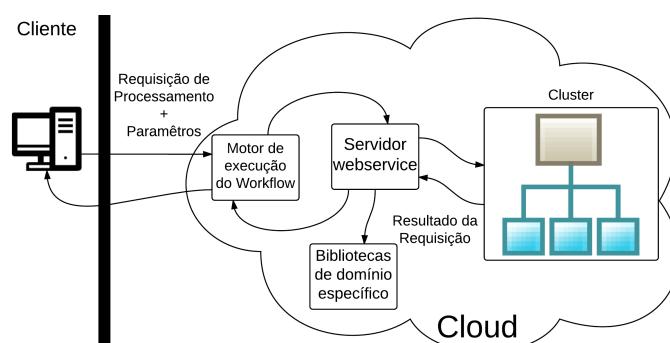


Figura 1. Arquitetura conceitual do ambiente proposto.

3.2. Aspectos de Implementação

Foi feita uma implementação da arquitetura proposta utilizando-se várias ferramentas e bibliotecas com propósitos complementares e as linguagens Java e Scala. A Figura 2 ilustra a organização dos componentes utilizados como camadas de processamento.

No nível mais externo, i.e. mais próximo ao usuário, encontra-se o sistema de gerenciamento de *workflows* científicos Taverna. A versão utilizada foi a 2.5. O Taverna¹ é um conjunto de ferramentas de workflows de código aberto² projetado para combinar *web services* distribuídos e/ou ferramentas para análises de *pipelines* complexas

¹<http://www.taverna.org.uk/>

²<https://github.com/taverna/>

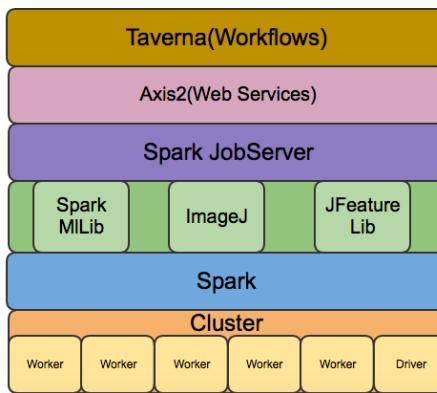


Figura 2. Uma visão geral dos componentes da implementação.

[Oinn et al. 2004]. O Taverna oferece uma interface gráfica fácil de ser manuseada que permite a criação de *workflows*. Além disso, o Taverna é responsável pelo fluxo de controle durante a execução das atividades do processo descrito, provendo a ligação entre atividades subsequentes e permitindo o acompanhamento da execução.

As atividades do *workflow* são executadas por meio de chamadas a *web services*. Na proposta, o Apache Axis2 (versão 1.7.0)³ foi utilizado como servidor das chamadas *web services*. O Axis2 é um *middleware*⁴ de código aberto⁵ para *web services*, mensagens SOAP e WSDL, que providencia abstrações e serviços que são utilizados em todos os aspectos da pilha que compõe os *web services*. Dentre as funcionalidades chaves providenciadas pelo Axis2 encontra-se o suporte ao estilo arquitetural REST (*Representational State Transfer*) [Perera et al. 2006], que ignora detalhes da implementação dos componentes e não armazena estado das comunicações entre mensagens e foi utilizado na implementação da proposta.

O motor de execução da arquitetura é o Apache Spark (versão 1.6.0)⁶. O Spark é um framework para processamento paralelo de grandes conjuntos de dados, que foi construído com foco em velocidade, facilidade de uso e análises sofisticadas [Zaharia et al. 2010]. Com um modelo de processamento simples, mas robusto e com alta tolerância à falhas e com bom desempenho, esse *framework* constitui uma das peças centrais da implementação, controlando a alocação de recursos e distribuindo o processamento para clusters de (potencialmente) milhares de máquinas. O gerenciamento dos dados no *cluster* de processamento utiliza os *Resilient Distributed Datasets* (RDD) do Spark. Um RDD é uma coleção somente-leitura de objetos que está particionada através de um conjunto de máquinas, e que pode ser reconstruída caso alguma dessas máquinas seja perdida. Na implementação atual, a solução utilizada para armazenamento em disco distribuído é baseada no NFS (*Network File System*) versão 4.

A ligação entre os *web services* e o Spark é feita por meio do Spark JobServer⁷. O Spark JobServer (versão 0.6.1) pode ser considerado um *middleware* entre uma aplicação

³<http://axis.apache.org/axis2/java/core/index.html>

⁴Middleware é toda aplicação que faz a mediação entre duas aplicações.

⁵<https://github.com/apache/axis2-java>

⁶<http://spark.apache.org/>

⁷<https://github.com/spark-jobserver/spark-jobserver>

e o Spark que providencia uma interface RESTful para submissão e gerenciamento de serviços de processamento (*jobs*). Na implementação, ao receber um chamada de *web service*, é executado um código relativo à chamada recebida que invoca um novo *job* no Spark JobServer, que é posteriormente submetida para execução pelo Spark no *cluster*.

Os algoritmos de processamento disponíveis são providos por bibliotecas de terceiros com execução no Spark. As funções de processamento de imagens são providas pela ImageJ [Schneider et al. 2012], que é uma biblioteca consolidada em matéria de processamento de imagens e possui suporte para todos os formatos de imagem mais utilizados, em particular o do padrão DICOM. Os algoritmos de extração de características de imagens são um subconjunto da biblioteca JFeatureLib [Graf 2015] e da biblioteca Lire [Lux and Chatzichristofis 2008]. Tratam-se de bibliotecas que trazem diversos extractores de características diferentes e que possuem uma integração muito natural com o ImageJ. Outra biblioteca utilizada na proposta é a MLLib [Meng et al. 2015], que, dentre outras funcionalidades, possui algoritmos de seleção de características com execução distribuída. A MLLib contém diversas implementações de algoritmos conhecidos adaptados para o modelo de processamento do Spark, com objetivo de suportar aplicações que utilizam a infraestrutura de processamento do Apache Spark para tarefas relacionadas ao aprendizado de máquina. Por fim, algumas outras funcionalidades foram codificadas para complementar as funcionalidades da arquitetura proposta, tais como funções de distância, métodos de avaliação de resultado (e.g. gráficos de precisão × revocação) e classes estruturais do sistema.

4. Exemplo de Uso da Arquitetura Proposta

Esta seção descreve um exemplo de uso da arquitetura proposta. O problema a ser resolvido é o seguinte. Para um dado domínio de imagens, deseja-se avaliar diferentes algoritmos de pré-processamento de imagens associados a diferentes extractores de características, a fim de escolher a melhor combinação. No caso, trata-se de um base de imagens de exames para a identificação de fraturas na coluna, obtidas no Hospital das Clínicas da Faculdade de Medicina de Ribeirão Preto (HCFMRP), da Universidade de São Paulo (USP) e classificadas por especialistas.

São duas as preocupações iniciais deste trabalho. A primeira é que a solução proveja um ambiente onde seja possível realizar análises que são comuns à aplicações de recuperação de imagens médicas por conteúdo. A segunda preocupação diz respeito à escalabilidade da solução. Ou seja, é esperado que exista um ganho de performance na medida em que mais máquinas são utilizadas para realizar o processamento que o *pipeline* definido exige. Assim, a seguir são apresentados a definição deste experimento e o resultado obtido, em termos de qualidade das respostas para as diferentes combinações, e também um teste de escalabilidade da proposta.

4.1. Análise de Combinações de Pré-processamento e Extractores de Características

Para a realização deste teste, foram selecionados na literatura cinco algoritmos de pré-processamento e cinco algoritmos de extração de características. Os algoritmos para pré-processamento utilizados neste experimento foram: realce de contraste, aguçamento, remoção de discrepância, binarização e segmentação com *watershed*. Os algoritmos de extração de características selecionados

foram: CEDD [Chatzichristofis and Boutilis 2008a], Histograma de Cores, FCTH [Chatzichristofis and Boutilis 2008b], Haralick [Haralick et al. 1973] e Momentos Estatísticos [Gletsos et al. 2003].

Para realizar as combinações entre os diferentes algoritmos de pré-processamento e de extração de características, foram definidos no Taverna *workflows* idênticos ao apresentado na Figura 3, que é para o extrator CEDD com realce de contraste. A entrada do *workflow* é o identificador do conjunto de imagens. A atividade *create_spark_context* é necessária para definir o contexto utilizado pelo Spark para a execução das tarefas. Na sequência, são executados o realce de contraste e a extração de características para todo o conjunto de imagens. Por fim, é gerado um gráfico de precisão e revocação para avaliar a qualidade da representação gerada como saída do *workflow*.

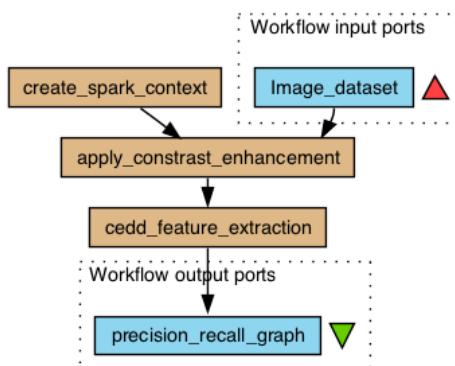


Figura 3. Workflow definido no Taverna para avaliação da combinação de realce de contraste e o extrator CEDD.

Foram gerados os *workflows* com as combinações, alterando-se apenas as atividades de pré-processamento e extração de características. Em outras palavras, cada imagem foi submetida a cada um dos cinco pré-processamentos, produzindo seis variações de imagens (uma original + cinco pré-processadas). Esse novo conjunto de imagens, que incluía as imagens modificadas, foi então utilizado como entrada para cada um dos cinco extractores de características. Dessa forma, ao final desse processo, foram gerados 30 vetores de características de cada imagem do conjunto inicial de imagens.

A Figura 4 apresenta o gráfico de precisão e revocação que é resultado da execução do *workflow* apresentado na Figura 3, quando este teve como conjunto de imagens de entrada as 171 imagens de coluna vertebral citadas anteriormente. Por se tratar de um conjunto de entrada relativamente pequeno, esse *workflow* foi executado no *cluster* com apenas um nó de processamento.

Na Figura 4 é possível visualizar a variação existente na qualidade de resultados recuperados na medida em que são utilizados diferentes pré-processamentos em imagens que tiveram suas características extraídas pelo algoritmo CEDD. A aplicação de realce de contraste possibilita algum ganho de qualidade em comparação a não fazer nenhum pré-processamento antes do CEDD (curva *identity* na figura). Por outro lado, a binarização, também utilizada para fazer a segmentação por *watershed*, degrada a qualidade das respostas. Gráficos semelhantes foram gerados para os demais extractores de características,

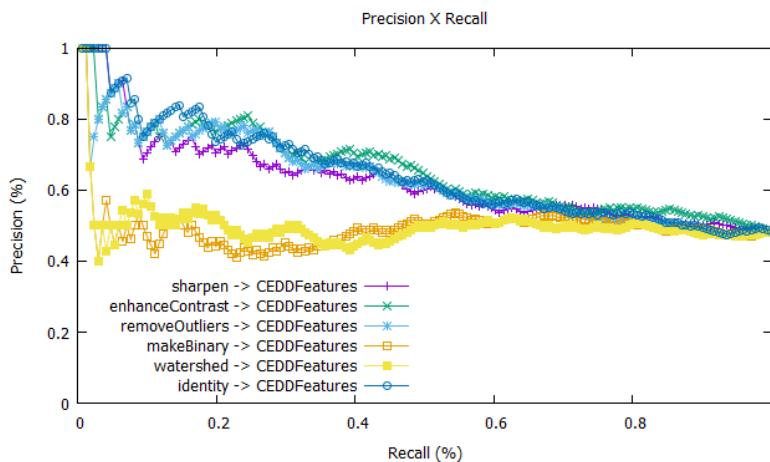


Figura 4. Precisão e revocação para algoritmo CEDD.

finalizando o estudo de caso. Observe-se que não foi o foco deste exemplo efetivamente explorar métodos para classificação de imagens de coluna quanto à presença de fraturas, mas mostrar a aplicabilidade da arquitetura proposta.

4.2. Teste de Escalabilidade

Para avaliar a escalabilidade da solução foram feitas quatro execuções dos workflows gerados para a análise descrita na seção anterior. Um teste com uma execução local, um teste com uma execução distribuída, mas apenas um nó trabalhador, um teste com três nós trabalhadores e um último teste com todos os cinco nós trabalhadores. Para este experimento foi utilizado um conjunto de dados com 22.000 imagens médicas de exames de pulmão, ocupando 12GB de espaço em disco. O *cluster* foi montado com cinco nós para processamento, sendo um deles responsável pelo gerenciamento de recursos. Todas as máquinas foram colocadas na mesma rede local e possuem configurações similares: quatro núcleos de processamento a 3.0GHz, com 2GB de memória RAM em cada nó disponibilizados para o Spark.

A Figura 5 apresenta a variação do tempo de execução de acordo com a quantidade de nós de processamento, em minutos, para o pré-processamento seguido da extração de características das 22.000 imagens médicas, para todas as combinações citadas na seção anterior. Nota-se que a execução no *cluster* com apenas um nó foi mais lenta do que a execução local, devido ao atraso proveniente do ambiente do *cluster*. Contudo, ao acrescentar-se nós ao *cluster*, o *speedup* foi praticamente linear, chegando a ser 4,35 vezes mais rápido do que a execução local quando foram utilizados os cinco nós.

Vale ressaltar que o teste local foi interrompido antes da fase de escrita dos resultados ser iniciada. Além disso, durante a execução distribuída com cinco nós trabalhadores, em certo momento houve uma falha em um dos nós, que foi desconectado do *cluster*. Contudo, o Spark provê mecanismos de recuperação de falha que contornaram esse problema, de forma que nenhum resultado foi perdido.

5. Conclusão e trabalhos futuros

Este artigo apresentou uma proposta de arquitetura para CBMIR baseada em *workflows* com execução distribuída. A implementação da proposta apresentada neste trabalho ainda

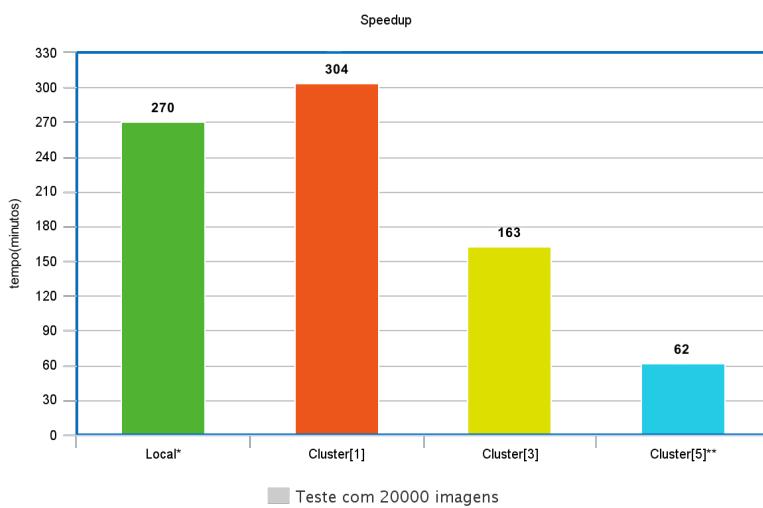


Figura 5. Impacto da distribuição do processamento no tempo de execução total.

é um trabalho em desenvolvimento. Contudo, algumas das características que eram requisiitos iniciais puderam ser atendidos, totalmente ou em parte, como o fato de a solução ser escalável para conjuntos de dados massivos, por meio do suporte do Spark, e a possibilidade de se fazer experimentos comuns à aplicações de CBMIR, a partir dos quais podem ser feitas análises que deem apoio à atividade médica.

Dentre os trabalhos futuros, podem ser citados uma melhora na interação dos usuários com os *workflows*, uma vez que na sua versão atual, ainda é exigido do usuário um processo de configuração dentro do ambiente de definição de *workflows* que diz respeito a detalhes do *cluster* que, idealmente, devem ser transparentes ao usuário. Outra possibilidade de trabalho futuro é em relação à adição de recursos que permitam a visualização de resultados intermediários dentro do *pipeline*, além da possibilidade de se visualizar gráficos gerados a partir dos dados de maneira automática, uma vez que hoje eles precisam ser gerados manualmente.

Referências

- Bedo, M. V. N., Ponciano-Silva, M., Kaster, D. S., Bugatti, P. H., Traina, A. J. M., and Traina Jr, C. (2012). Higiia: A Perceptual Medical CBIR System Applied to Mammography Classification. In *Demo and Applications Session of the XXVII Brazilian Symposium on Databases (SBBD)*, pages 13–18, São Paulo, SP.
- Burak Akgül, C., Rubin, D. L., Napel, S., Beaulieu, C. F., Greenspan, H., and Acar, B. (2011). Content-based image retrieval in radiology: Current status and future directions. *Journal of Digital Imaging*, 24:208–222.
- Chatzichristofis, S. A. and Boutalis, Y. S. (2008a). CEDD: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. pages 312–322.
- Chatzichristofis, S. A. and Boutalis, Y. S. (2008b). FCTH: Fuzzy Color and Texture Histogram - A Low Level Feature for Accurate Image Retrieval. In *2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services*, pages 191–196. IEEE.

- Deserno, T. M., Antani, S., and Long, R. (2009). Ontology of gaps in content-based image retrieval. *Journal of Digital Imaging*, 22(2):202–215.
- Gletsos, M., Mougiakakou, S., Matsopoulos, G., Nikita, K., Nikita, A., and Kelekis, D. (2003). A computer-aided diagnostic system to characterize CT focal liver lesions: design and optimization of a neural network classifier. *IEEE Transactions on Information Technology in Biomedicine*, 7(3):153–162.
- Grace, R. K., Manimegalai, R., and Kumar, S. S. (2014). Medical image retrieval system in grid using hadoop framework. In *Proceedings - 2014 International Conference on Computational Science and Computational Intelligence, CSCI 2014*, volume 1, pages 144–148. IEEE.
- Graf, F. (2015). Jfeaturelib v1.6.3.
- Haralick, R. M., Shanmugam, K., and Dinstein, I. H. (1973). Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, (6):610–621.
- Jai-Andalousi, S., Elabdouli, A., Chaffai, A., Madrane, N., and Sekkaki, A. (2013). Medical content based image retrieval by using the Hadoop framework. In *Ict 2013*, pages 1–5. IEEE.
- Larobina, M. and Murino, L. (2014). Medical image file formats. *Journal of digital imaging*, 27(2):200–206.
- Lux, M. and Chatzichristofis, S. A. (2008). Lire: lucene image retrieval: an extensible java cbir library. In *Proceedings of the 16th ACM international conference on Multimedia*, pages 1085–1088. ACM.
- Meng, X., Bradley, J. K., Yavuz, B., Sparks, E. R., Venkataraman, S., Liu, D., Freeman, J., Tsai, D. B., Amde, M., Owen, S., Xin, D., Xin, R., Franklin, M. J., Zadeh, R., Zaharia, M., and Talwalkar, A. (2015). Mllib: Machine learning in apache spark. *CoRR*, abs/1505.06807.
- Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M. R., Wipat, A., et al. (2004). Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054.
- Perera, S., Herath, C., Ekanayake, J., Chinthaka, E., Ranabahu, A., Jayasinghe, D., Weerawarana, S., and Daniels, G. (2006). Axis2, middleware for next generation Web Services. *Proceedings - ICWS 2006: 2006 IEEE International Conference on Web Services*, pages 831–840.
- Schneider, C. A., Rasband, W. S., and Eliceiri, K. W. (2012). NIH Image to ImageJ: 25 years of image analysis. *Nature Methods*, 9(7):671–675.
- Sridharan, R. (2015). *Visualization and Analysis of Large Medical Image Collections Using Pipelines*. PhD thesis, Massachusetts Institute of Technology.
- Wolf, I. (2010). Toolkits and software for developing biomedical image processing and analysis applications. In *Biomedical Image Processing*, pages 521–544. Springer Berlin Heidelberg.
- Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. (2010). Spark : Cluster Computing with Working Sets. *HotCloud'10 Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, 10:10.

aper:152933_1

XplNet – Análise exploratória aplicada à redes complexas

Luiz Gomes-Jr¹, Nádia P. Kozievitch¹, André Santanchè²

¹Departamento de Informática
Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba – PR – Brasil

²Instituto de Computação
Universidade Estadual de Campinas (UNICAMP)
Campinas, SP – Brasil

{gomesjr,nadiap}@dainf.ct.utfpr.edu.br, santanche@ic.unicamp.br

Resumo. Análise exploratória se tornou uma ferramenta essencial em diversos processos de análise de dados. No contexto de redes complexas, a alta dimensionalidade dos dados e diversidade dos relacionamentos dificulta interações exploratórias. Este desafio é ainda maior em interações que combinem navegação, seleção e análise dos dados – uma combinação típica nas investigações. Inspirados na interação oferecida pelos mecanismos de OLAP no mundo relacional, nosso objetivo neste trabalho é propor mecanismos equivalentes para análise de redes complexas. Nossa proposta inclui um modelo de operações exploratórias, respectivos elementos visuais, e arquitetura do sistema. A proposta está baseada na Beta álgebra e mecanismos de consulta e gerenciamento desenvolvidos no contexto do CDMS (Complex Data Management System). Ilustramos nossa proposta com um caso de uso baseado em dados abertos.

1. Introdução

Eficiência e flexibilidade na análise de dados é um fator decisivo em diversos contextos no mundo moderno. Inteligência de negócios, eScience, BigData e Redes Complexas são exemplos de áreas dependentes de uma alta capacidade de análise de dados. As estratégias de análise de dados podem ser divididas em duas categorias: análise baseada em modelos e análise exploratória. Análise baseada em modelos é direcionada para casos onde o objetivo da análise é bem definido, estabelecendo como desafio a definição e implementação de um modelo adequado. Análise exploratória, por sua vez, é indicada para todos os casos onde não há informação suficiente para se propor um modelo para o problema. A análise exploratória busca construir modelos e respectivas hipóteses a partir de um processo de interação com os dados e descoberta de informação.

Análise exploratória de dados foi promovida pelo matemático John Tukey na década de 70 [Tukey 1977] e obteve grande sucesso em diversos contextos [Andrienko and Andrienko 2006, Ibragimov et al. 2014]. Nos dias de hoje, as técnicas são aplicadas em mineração de dados e big data. OLAP (Online Analytical Processing) é uma manifestação das técnicas de análise exploratória para dados relacionais. O cubo OLAP se tornou um grande expoente na área devido à forma simples e efetiva de representar e manipular dados multidimensionais.

Existe, contudo, um desafio aberto relacionado à aplicação de análise exploratória sobre dados altamente interconectados, conhecidos como dados de rede ou grafos. Tais dados estão se tornando cada vez importantes e comuns em diversas aplicações.

Redes sociais e buscadores da web são exemplos de uso intensivo de dados e análise de redes. As áreas de aplicação estão se expandindo à medida que mais dados interconectados são gerados e as necessidades de análise se ampliam. A área de redes complexas (também conhecida como ciência de redes) foi desenvolvida em resposta a estas demandas e se desenvolveu rapidamente nos últimos anos [da F. Costa et al. 2007, Newman 2003]. Intrinsecamente multidisciplinar, a área investiga as características e o efeito da topologia dos relacionamentos em redes. Por exemplo, nós com alto grau de ligação com outros nós são chamados de *hubs* e demonstram características especiais sejam em redes sociais como o Facebook, ou em redes artificiais como a rede de roteadores da Internet [da F. Costa et al. 2011].

Análise de redes complexas é uma clara candidata à utilização de técnicas de análise exploratória. Porém, a complexidade e alta dimensionalidade dos dados tornam as técnicas tradicionais inadequadas. Navegação ou seleção bem como agregação dos dados são elementos básicos da análise exploratória, porém difíceis de se definir num contexto de redes. Além disso, as estruturas atuais de armazenamento e processamento de dados não disponibilizam interfaces adequadas a este tipo de análise.

Este artigo propõe mecanismos para análise exploratória em redes complexas, tomando como base modelos e componentes desenvolvidos nos últimos anos por co-autores no contexto do CDMS (Complex Data Management System). As principais contribuições do artigo são:

- Descrição do workflow que buscamos oferecer aos analistas de redes complexas (Seção 3.1).
- Especificação inicial do modelo de interação exploratória (Seção 3.2).
- Especificação inicial dos elementos visuais de suporte ao modelo de interação (Seção 3.3).
- Organização da arquitetura do sistema (Seção 3.4).
- Descrição de um caso de uso baseado em dados abertos (Seção 4).

2. Trabalhos relacionados e fundamentos

Análise exploratória surgiu como um ramo da análise estatística, proposta por John Tukey na década de 1970 [Tukey 1977]. Análise exploratória se foca nos casos onde o conhecimento a priori dos dados é limitado. As técnicas contrastam com a abordagem tradicional de teste de hipótese, onde as variáveis envolvidas e as questões da análise são bem compreendidas, permitindo a definição e verificação de modelos estatísticos.

Numa análise exploratória de dados, o profissional utiliza de ferramentas capazes de selecionar e agrregar os dados de interesse e, principalmente, se apóia em elementos visuais para construir uma melhor compreensão dos dados. Entre ferramentas e técnicas comuns na área estão a análise de distribuição das variáveis (e.g. histogramas), *scatter plots*, análise de componente principal, análise de clusters, etc.

No campo dos dados relacionais, análise OLAP (OnLine Analytical Processing) é um caso bem sucedido de análise exploratória, tanto no âmbito de pesquisa quanto na

indústria. OLAP permite análise exploratória de dados multidimensionais através do cubo OLAP, uma estrutura que permite a aplicação de operações de manipulação e agregação dos dados.

As operações básicas suportadas por OLAP são fatiar (slice), subcubo (dice), agregação (roll-up) e detalhamento (drill-down). Estas operações manipulam o cubo multidimensional permitindo que a análise se desenvolva à medida que o analista encontra padrões interessantes nos dados. Nossa objetivo é oferecer uma interação equivalente sobre redes complexas.

Pesquisadores no campo de redes complexas têm desenvolvido uma grande gama de modelos e algoritmos para analisar a dinâmica das redes. As técnicas utilizam diversas estratégias para derivar métricas para a análise, como contagem de caminhos, fatorização de matrizes, análise estrutural, etc [da F. Costa et al. 2011].

Pesquisadores de redes complexas se apoiam tipicamente numa combinação de scrips de código, análise gráfica, e software matemático para realizar os estudos. Apesar da importância e do alcance da área, ainda não há uma padronização de técnicas, abordagens ou softwares nas análises.

Redes complexas são frequentemente representadas como grafos. Existem diversas linguagens e sistemas de bancos de dados disponíveis para lidar com grafos [Wood 2012, Angles and Gutierrez 2008]. Embora as opções disponíveis ofereçam vantagens em termos de armazenamento e gerenciamento de dados, elas não oferecem mecanismos capazes de capturar as propriedades emergentes da redes. Por exemplo, elas não permitem que usuários selecionem uma subporção do grafo e apliquem uma análise (e.g. PageRank) baseada em critérios específicos.

Em termos de escalabilidade, existem diversos frameworks para armazenamento e processamento de grandes grafos. São exemplos desta categoria o Pregel do Google [Malewicz et al. 2010] e o GraphLab [Low et al. 2012], que permitem que o processamento seja distribuído em múltiplos computadores. As complexidades de implementação e configuração destes sistemas e a curva de aprendizado dos modelos envolvidos restringem suas aplicações à grandes empresas. Bancos de dados distribuídos como o Titan¹ também estão sendo desenvolvidos, suportando um workflow mais tradicional de gerenciamento de dados, mas ainda não oferecendo nativamente as operações necessárias para análise de redes complexas.

Vizualização é um aspecto importante na análise de redes [Herman et al. 2000, Pavlopoulos et al. 2008]. Existem diversas ferramentas para visualização e análise de redes. O processamento é, no entanto, executado globalmente e os efeitos são igualmente aplicados no grafo inteiro. Embora por vezes possível, o tipo de interação exploratória que estamos propondo não é suportado nativamente nas ferramentas.

2.1. Álgebra Beta

Um dos elementos fundamentais da nossa proposta, que permitirá explorar interativamente o grafo para análise, é a possibilidade de transformar as interações do analista pela interface em operações que possam ser compreendidas, otimizadas e executadas pelo sis-

¹<http://thinkaurelius.github.io/titan/>

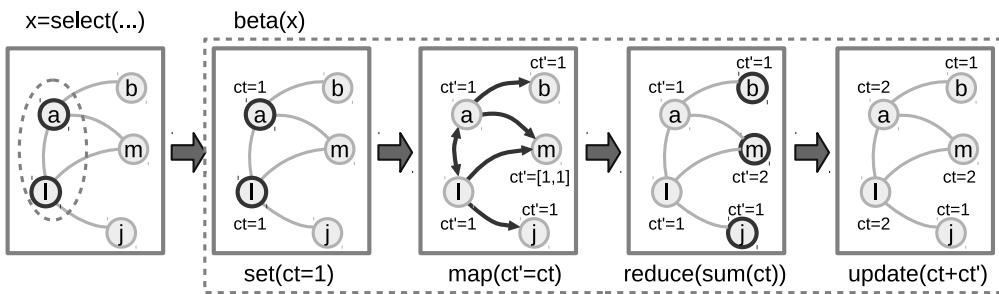


Figura 1. Interpretação simplificada em grafo de uma iteração do operador Beta.

tema de gerenciamento de dados. Em passos anteriores desta pesquisa [Gomes-Jr 2015], desenvolvemos a álgebra Beta como um modelo capaz de desempenhar este papel.

A álgebra Beta estende a álgebra relacional com o operador Beta, responsável por operações de agregação no grafo. O modelo relacional permite uma representação direta de grafos (a definição matemática de um grafo é relacional) e também facilita análises independentes da topologia (como contagens e agregações simples). O operador Beta é especializado em agregações ao longo das arestas, permitindo análises de caminhos (e.g. distâncias e conectividade) e análises globais de convergência (e.g. PageRank, HITS).

Resumidamente, o operador Beta executa joins recursivos sobre a tabela de arestas e aplica, simultaneamente, sub-operações de agregação. As sub-operações, passadas como parâmetro para o operador, são *set*, *map*, *reduce*, e *update*. Elas determinam os novos atributos calculados ao longo do atravessamento do grafo. A Figura 1 mostra uma interpretação em grafo dos operadores relacionais executados para a contagem dos caminhos entre dois nós iniciais (*a*, *l*) e os demais. O sub-operador *set* determina os valores iniciais do atributo *ct*, que então é transferido para nós vizinhos pelo sub-operador *map*. O sub-operador *reduce* realiza uma agregação nos nós de valores comuns e, finalmente, o sub-operador *update* atualiza os valores da iteração anterior do operador Beta. O processo continua até que o critério de parada seja satisfeito (e.g. número determinado de passos ou convergência de valores). Uma definição detalhada, com descrição dos parâmetros e exemplos do operador Beta pode ser encontrada em [Gomes-Jr 2015].

A álgebra Beta é uma opção adequada para implementação das operações de análise propostas neste trabalho porque: (i) é capaz de representar simultaneamente operações de atravessamento e agregação ao longo dos caminhos (incluindo medidas que requerem convergência como PageRank); (ii) é baseada em álgebra relacional, permitindo operações de manipulação e agregação das tabelas retornadas; (iii) oferece oportunidades de otimização transparente das consultas.

3. Análise exploratória de redes complexas

Esta seção descreve os elementos básicos da nossa proposta de interação, cobrindo o workflow desejado, aspectos de modelagem e arquitetura.

3.1. Workflow

Baseado nas bem-sucedidas tecnologias OLAP, nosso objetivo é oferecer um framework que possibilite análise exploratória de redes complexas. A Figura 2 ilustra nossa visão da

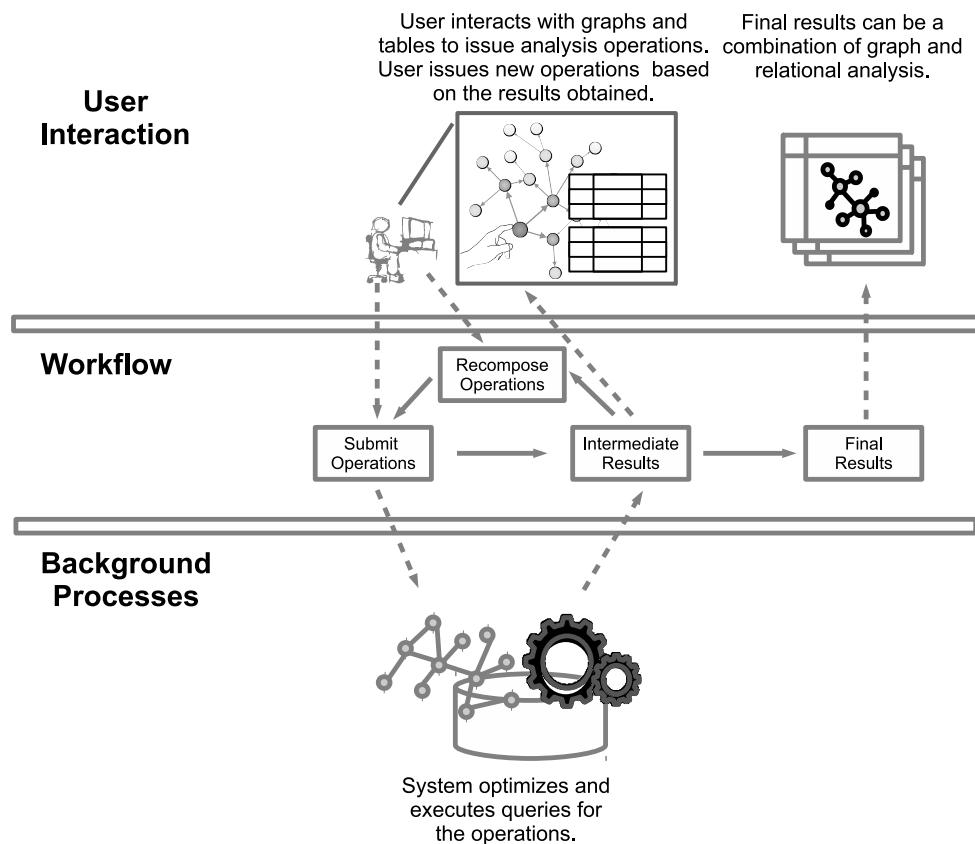


Figura 2. Workflow de análise exploratória de redes.

interação entre analistas e o framework. A interface com o usuário oferece visualização do subgrafo selecionado e detalhamento dos dados relacionais (propriedades dos nós e arestas).

O usuário interage com o grafo através de operações de navegação e análise (equivalentes aos *drill downs* e *roll ups* do OLAP) e recebe o feedback visual do grafo ativo e detalhamento das propriedades (atributos e valores calculados na análise). Baseado no feedback, o usuário formula novas operações de análise, repetindo o ciclo até que esteja satisfeito com o resultado.

Questões de carga e gerenciamento de dados neste contexto são tratadas em [Gomes-Jr 2015].

3.2. Modelo de consultas exploratórias

Um dos maiores desafios para análise exploratória em redes é a definição de um modelo que integre naturalmente aspectos de navegação e análise em grafos. Ao contrário do que se vê num cenário OLAP, grafos não têm esquema nem dimensões definidas, o que dificulta a especificação de operações que combinem simultaneamente navegação e agregação.

Argumentamos, porém, que a Beta álgebra é uma candidata adequada para fazer esta integração, uma vez que é capaz de representar operações de navegação bem como operações de agregação ao longo do grafo. Nossa objetivo, portanto, é especificar nosso modelo de consultas exploratórias com base na álgebra Beta.

O modelo de consultas exploratórias proposto neste trabalho é composto de três operações: *Select*, *Traverse* e *Analyze*. *Select* é responsável por selecionar nós a partir de um critério booleano ou nós especificados na interface. *Select* é equivalente ao operador relacional, neste contexto usado para iniciar e direcionar as tarefas de análise.

Traverse é a operação responsável pela navegação no grafo. Uma operação de atravessamento pode também especificar agregações para serem executadas ao longo da navegação. Por exemplo, o analista pode executar uma operação de atravessamento de arestas e simultaneamente calcular a distância para chegada nos nós de destino. Em termos da álgebra Beta, uma operação de atravessamento é equivalente à especificação do operador Beta com o suboperador *reduce* agregando somente sobre o *id* do nó de origem.

Analyze realiza operações de medição de redes complexas sobre os nós ativos na interface. Exemplos deste tipo de análise são algoritmos que calculam a centralidade dos nós, como o PageRank. Na álgebra Beta, este tipo de operação é equivalente ao operador Beta onde *reduce* agrupa sobre os *ids* dos nós de origem e destino, *update* agrupa valores calculados na última iteração com valores da anterior e, por fim, um critério de parada baseado em convergência é especificado.

3.3. Elementos visuais

A definição do modelo de consultas exploratórias tem pouco valor se não for acompanhada de elementos visuais capazes de auxiliar o especialista na especificação das análises e navegações bem como apresentar os resultados de forma intuitiva.

Especificamos aqui cinco elementos básicos para suportar a interação desejada:

- Selecionar: elementos visuais devem permitir que o usuário selecione uma subporção do grafo. Nós selecionados devem ser claramente marcados como ativos.
- Expandir/retrair: nas operações de *traverse* a navegação deve ser representada por elementos distintos representando os dados antes e depois da aplicação da operação. Por exemplo, na Figura 4 (passo 4), os nós ativos antes da operação são representados agrupados no centro do grafo expandido. Retrair é equivalente à operação inversa (uma espécie de *undo*).
- Centralizar: nós selecionados podem ser escolhidos como foco de análises futuras. Os elementos visuais devem dar destaque a estes nós (e.g. centralizando-os).
- Clusterizar: nós podem ser agrupados de acordo com variáveis qualitativas (tipo, região, faixa etária, etc.) ou por operações de análise (identificação não supervisionada de classes). A interface deve oferecer elementos para distinguir grupos (e.g. cores) e arranjá-los no espaço.
- Transformar: o analista pode optar por redefinir o grafo ativo, por exemplo, modificando o conjunto de arestas de interesse. A interface deve permitir esta seleção e atualizar o grafo ativo.

Este é um conjunto básico de elementos para permitir a interação exploratória com redes complexas. Um desafio extra é o fato de que as análises frequentemente se baseiam em grafos grandes demais para exibição na tela. Portanto, estes elementos visuais devem oferecer mecanismos para resumo dos dados indicando claramente que existem elementos omitidos e possibilitando a expansão destes.

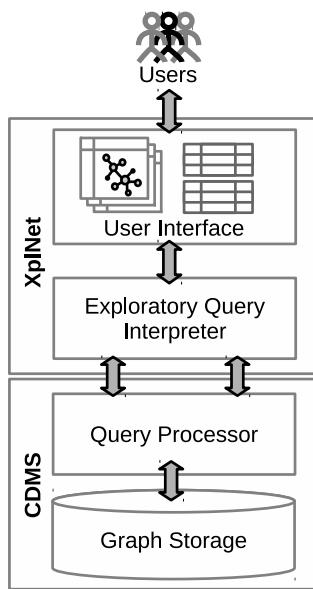


Figura 3. Arquitetura proposta para o XplNet

3.4. Arquitetura

A arquitetura do XplNet está divida nos seguintes componentes principais: (i) Interface com o usuário, (ii) Interpretador de consultas exploratórias, (iii) Processador de consultas, (iv) Gerenciador de armazenamento. Processamento de consultas (e modelos relacionados) foi tratado em passos anteriores desta pesquisa [Gomes-Jr 2015].

A interface com o usuário é o único ponto de interação entre profissionais e o sistema. Ela deve permitir a execução das operações do modelo de consultas de uma maneira intuitiva, oferecendo feedback visual das operações e simplificando a alteração de parâmetros da consulta.

O interpretador de consultas exploratórias recebe as chamadas de operações da interface e as traduz para o modelo de consultas do processador. Nesta proposta as operações são traduzidas para a álgebra Beta.

O processador de consultas recebe a consulta em álgebra Beta, a transforma numa representação interna e executa procedimentos de análise e otimização para gerar um plano de execução.

Por fim, o gerenciador de armazenamento é responsável por obter os dados armazenados (possivelmente de forma distribuída) e encaminhá-los para o processador de consultas.

4. Caso de uso

Para ilustrar o uso do framework proposto em uma tarefa de análise, nos basearemos num cenário de uso de dados abertos no contexto de cidades inteligentes. Nos basearemos nos dados de registro de solicitações do cidadão, a *Central de Atendimento e Informações 156*, da cidade de Curitiba-PR². O objetivo do cenário é determinar e compreender as principais questões levantadas pelos usuários do serviço.

²<http://www.curitiba.pr.gov.br/dadosabertos/>

| Nome do campo | Descrição | Tipo |
|---------------|--|----------|
| SOLICITACAO | Nº da solicitação efetuada na Central de Atendimento 156 | integer |
| TIPO | Tipo da solicitação. Ex: solicitação, elogio, reclamação,etc.. | varchar |
| ORGAO | Órgão em que cadastrador está lotado | varchar |
| DATA | Data da criação da solicitação | datetime |
| HORARIO | Hora da criação da solicitação | datetime |
| ASSUNTO | Assunto ao qual a solicitação se refere | varchar |
| SUBDIVISAO | Subdivisão do assunto ao qual a solicitação se refere | varchar |
| DESCRICAO | Descrição da solicitação | varchar |

Tabela 1. Descrição dos primeiros campos dos dados do serviço 156.

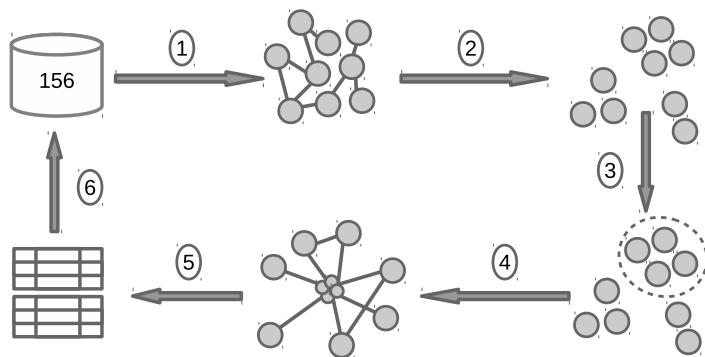


Figura 4. Cenário de análise simplificado.

A Tabela 1 mostra os oito primeiros campos dos dados disponibilizados pela prefeitura. O grafo representando estes dados pode ser uma derivação direta do esquema relacional normalizado, com os registros das tabelas definidos como nós e as chaves estrangeiras representando arestas. Este tipo de transformação de modelo retém a regularidade do modelo relacional e não utiliza toda a flexibilidade proposta neste trabalho. Para tornar o grafo mais complexo e a análise mais interessante, incrementamos o grafo com arestas formadas a partir do conteúdo textual do campo descrição. Neste cenário, cada palavra (potencialmente excluindo palavras pouco relevantes como artigos e preposições) se torna um nó no grafo e registros contendo esta palavra são ligados ao nó criado. Por exemplo, solicitações contendo a palavra “ruído” no campo descrição são ligadas ao nó que representa a palavra. Detalhes sobre a construção do grafo estão fora do escopo deste artigo, mas podem ser obtidos em [Gomes-Jr and Santanchè 2014].

A Figura 4 mostra uma esquemática simplificada do nosso cenário de análise. No passo 1 o analista carrega os dados do grafo na interface (alternativamente, o usuário pode carregar um subgrafo usando operações de seleção). O passo 2 representa uma tarefa de clusterização, realizada pela operação *analyze*. Por exemplo, os dados podem ser clusterizados com base nos relacionamentos formados por palavras em comum no

campo descrição. Este tipo de clusterização agregaria registros que contêm palavras em comum, organizando os registros em tópicos latentes [Mihalcea and Radev 2011]. No passo 3, aplicando uma operação *select*, o analista seleciona um dos agrupamentos para exploração.

No passo 4, o analista decide aplicar uma operação de *traverse* associada à uma métrica de cálculo de relevância (detalhes da representação deste tipo de análise na álgebra Beta podem ser obtidos em [Gomes-Jr 2015]). A expansão é especificada sobre os relacionamentos entre registros e palavras contidas. A representação visual da operação mostra os nós de origem aglomerados no centro e os nós alcançados em destaque ao redor. O resultado da operação também pode ser visualizado em forma de tabela, que ranqueia as palavras de acordo com o valor obtido pela métrica de relevância, por exemplo, tuplas como (“ruído”, 14.2) e (“engarrafamento”, 11.3). A informação obtida pode ser armazenada no banco para outras análises exploratórias, como para avaliação de áreas críticas associadas aos principais tópicos identificados.

No contexto atual, este simples caso de uso demandaria interações mais complexas entre diferentes ferramentas. Nossa objetivo é fazer com que tarefas como esta sejam tratadas de forma natural em um ambiente de análise intuitivo.

5. Conclusão

Apresentamos neste artigo nossa proposta de sistema para análise exploratória em redes complexas. Nos baseamos na álgebra Beta e mecanismos de gerenciamento de dados do CDMS (Complex Data Management System) para construir um modelo de interação e seus respectivos elementos visuais. Apresentamos também a arquitetura do sistema e um caso de uso baseado em dados abertos.

Como trabalhos futuros destacamos a definição formal do modelo de interação, design e testes dos elementos visuais, e implementação completa da arquitetura. Nossa objetivo é aplicar o sistema e as técnicas desenvolvidas em diversas áreas, como cidade inteligentes, redes sociais, e-Science, etc.

Referências

- Andrienko, N. V. and Andrienko, G. L. (2006). *Exploratory analysis of spatial and temporal data - a systematic approach*. Springer.
- Angles, R. and Gutierrez, C. (2008). Survey of graph database models. *ACM Computing Surveys*, 40(1):1–39.
- da F. Costa, L., Oliveira Jr, O., Travieso, G., Rodrigues, F., Boas, P., Antiqueira, L., Viana, M., and Rocha, L. (2011). Analyzing and modeling real-world phenomena with complex networks: A survey of applications. *Advances in Physics*, 60:329–412.
- da F. Costa, L., Rodrigues, F. A., Travieso, G., and Boas, P. R. V. (2007). Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1):167–242.
- Gomes-Jr, L. (2015). *Querying and managing complex networks*. PhD thesis, UNICAMP.
- Gomes-Jr, L. and Santanchè, A. (2014). The Web Within: leveraging Web standards and graph analysis to enable application-level integration of institutional data. *Transactions on Large Scale Data and Knowledge Centered Systems*.

- Herman, Melançon, G., and Marshall, M. S. (2000). Graph visualization and navigation in information visualization: A survey. In *IEEE Transactions on Visualization and Computer Graphics*, volume 6 (1), pages 24–43. IEEE Computer Society.
- Ibragimov, D., Hose, K., Pedersen, T. B., and Zimányi, E. (2014). Towards exploratory OLAP over linked open data - A case study. In Castellanos, M., Dayal, U., Pedersen, T. B., and Tatbul, N., editors, *BIRTE*, volume 206 of *Lecture Notes in Business Information Processing*, pages 114–132. Springer.
- Low, Y., Bickson, D., Gonzalez, J., Guestrin, C., Kyrola, A., and Hellerstein, J. M. (2012). Distributed graphlab: A framework for machine learning and data mining in the cloud. *Proc. VLDB Endow.*, 5:716–727.
- Malewicz, G., Austern, M. H., Bik, A. J. C., Dehnert, J. C., Horn, I., Leiser, N., and Czajkowski, G. (2010). Pregel: a system for large-scale graph processing. In *SIGMOD*.
- Mihalcea, R. and Radev, D. R. (2011). *Graph-based Natural Language Processing and Information Retrieval*. Cambridge University Press.
- Newman, M. (2003). The structure and function of complex networks. *SIREV: SIAM Review*, 45.
- Pavlopoulos, G. A., Wegener, A.-L., and Schneider, R. (2008). A survey of visualization tools for biological network analysis. *BioData Mining*, 1.
- Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley (Reading MA).
- Wood, P. T. (2012). Query languages for graph databases. *SIGMOD Record*, 41(1):50–60.

Artigos de Aplicações e Experiências

| | |
|--|-----|
| eC {^e}ncias | |
| Ciência de dados: Explorando Três Décadas de Evolução da Atividade Econômica em Curitiba | 139 |
| <i>Josana Rosa (Universidade Tecnológica Federal do Paraná), Thiago Henrique Silva (Universidade Tecnológica Federal do Paraná), Nádia Kozievitch (Universidade Tecnológica Federal do Paraná), Artur Ziviani (Laboratório Nacional de Computação Científica)</i> | |
| DataUSP: Conjunto de Serviços Analíticos para Apoio à Tomada de Decisões em uma Instituição de Ensino Superior | 143 |
| <i>Marino Hilário Catarino (Universidade de São Paulo), Bruno Padilha (Universidade de São Paulo), João Eduardo Ferreira (Universidade de São Paulo)</i> | |
| Dedup: um Aplicativo para Deduplicação de Contatos em Dispositivos Android | 147 |
| <i>Rafael F. Pinheiro (Universidade Federal do Rio Grande), Rafael Machado (Universidade Federal do Rio Grande), Eliza A. Nunes (Universidade Federal do Rio Grande), Eduardo N. Borges (Universidade Federal do Rio Grande)</i> | |
| RelationalToGraph: Migração Automática de Modelos Relacionais para Modelos Orientados a Grafos | 151 |
| <i>Gabriel Zessin (Universidade Estadual de Maringá), Edson Oliveira Jr (Universidade Estadual de Maringá)</i> | |
| Uma Proposta para Apresentar a Computação/Banco de Dados no Ensino Médio para o PÚblico Feminino | 155 |
| <i>Juan J. Rodriguez (Universidade Tecnológica Federal do Paraná), Nádia Kozievitch (Universidade Tecnológica Federal do Paraná), Sílvia A. Bim (Universidade Tecnológica Federal do Paraná), Mariangela de O. G. Setti (Universidade Tecnológica Federal do Paraná), Maria C. F. P. Emer (Universidade Tecnológica Federal do Paraná), Marília A. Amaral (Universidade Tecnológica Federal do Paraná)</i> | |
| xml2arff: Uma Ferramenta Automatizada de Extração de Dados em Arquivos XML para Data Science com Weka e R | 159 |
| <i>Gláucio R. Vivian (Universidade de Passo Fundo), Cristiano R. Cervi (Universidade de Passo Fundo)</i> | |

aper:152929_1

Ciência de dados: Explorando três décadas de evolução da atividade econômica em Curitiba

Josana Rosa¹, Thiago Henrique Silva¹, Nádia P. Kozievitch¹, Artur Ziviani²

¹ Universidade Tecnológica Federal do Paraná (UTFPR), Curitiba, PR – Brasil

²Laboratório Nacional de Computação Científica (LNCC), Petrópolis, RJ – Brasil

josanarosa@gmail.com, {thiagohs,nadiap}@utfpr.edu.br, ziviani@lncc.br

Resumo. Este artigo apresenta uma pesquisa em andamento que explora dados abertos sobre alvarás na cidade de Curitiba ao longo de mais de três décadas. Em particular, temos como objetivo fazer uma análise de três bairros, tirando proveito de técnicas de geoprocessamento e ciência de dados. Como resultado deste estudo apresentamos uma análise preliminar da evolução econômica da cidade.

Abstract. This paper presents an on-going research that explores open data about trade permits from the city of Curitiba over more than three decades. In particular, we aim at performing an analysis of three districts, taking advantage of GIS techniques and data science. As a result of this study we present a preliminary analysis of economic development city.

1. Introdução

Com o acelerado desenvolvimento urbano observado nas grandes cidades brasileiras, o planejamento urbano necessita de atualizações regulares da sua base cartográfica e eficientes técnicas de processamento e análise de dados. Nesse contexto, os Sistemas de Informações Geográficas (SIGs) em conjunto com modelos e técnicas computacionais têm sido aplicados na área de planejamento urbano, facilitando o trabalho de análises geográficas com o processamento de dados, auxiliando no gerenciamento e nas tomadas de decisões eficientes [Jat et al. 2008, Triantakonstantis and Mountrakis 2012].

Em particular, um grupo de cidades¹ definiu metas ambiciosas para melhorar a qualidade de vida urbana e proteger o meio ambiente. Curitiba desenvolveu e implementou corredores de transporte de massa, tornando-se um modelo de cidade sustentável com base em conceitos urbanísticos que moldaram a paisagem da cidade. Além disso, a cidade tem trabalhado com o conceito de dados abertos através da Prefeitura Municipal de Curitiba (PMC)² e do Instituto de Pesquisa e Planejamento Urbano de Curitiba (IPPUC)³.

Dentro deste contexto, este artigo apresenta estudos iniciais da evolução da atividade econômica ao longo de mais de três décadas em Curitiba com base em dados de alvarás concedidos na cidade. A Seção 2 apresenta os dados e ferramentas usados. A Seção 3 mostra alguns resultados preliminares acerca da evolução da atividade econômica. Finalmente, a Seção 4 apresenta a conclusão e trabalhos futuros deste projeto em andamento.

¹<http://www.c40.org> – Visitado em 02/03/2016.

²<http://www.curitiba.pr.gov.br/DADOSABERTOS/> – Visitado em 15/05/2015.

³<http://www.ippuc.org.br> – Visitado em 15/05/2015.

2. Dados e Ferramentas

Inicialmente, um conjunto de dados históricos de alvarás fornecido pela PMC foi selecionado para os bairros Centro (24.363 registros), Batel (5.099 registros) e Tatuquara (2.662 registros). Os dados estavam entre o período de 01 de Janeiro de 1980 a 31 de dezembro de 2013. Aproximadamente 0,05% dos registros apresentaram problemas para a representação direta do dado geocodificado, necessitando intervenção manual para a correção de latitude e longitude. Os dados possuíam somente as datas de criação, não permitindo saber quando o comércio foi fechado. As atividades econômicas, inicialmente catalogadas em 2.977 tipos (como Restaurante Dançante, Restaurante Pizzaria, etc.), foram agregadas em 68 tipos (como cartório, restaurante, banca, construtora, etc.). Em uma segunda etapa, foram integrados os dados de arruamento fornecidos pelo IPPUC (39.948 registros). Os dados do IPPUC e PMC foram importados para um servidor PostGIS, onde tablespaces, índices e esquemas específicos foram criados. Posteriormente, para a integração e visualização com as outras fontes (*GoogleMaps* e *OpenStreetMaps*⁴), foi utilizado o software QGIS⁵. A análise ilustrou que as fontes continham algumas diferenças, como no número total de ruas consideradas, nomes, entre outros. A Figura 1 apresenta a visualização dos bairros Centro, Batel e Tatuquara, entre os anos de 1980-1985 (esquerda) e sua totalização até 2013 (direita). Áreas mais escuras indicam a sobreposição de alvarás, indicando regiões com maiores concentrações históricas de comércio.

3. Evolução da Atividade Econômica

Em uma análise preliminar, nós nos concentrarmos em estudar a evolução temporal da concessão de alvarás na cidade de Curitiba do período de 1980 e 2013. Nesse estudo, avaliamos não somente a quantidade de alvarás concedidos ao longo do tempo, mas também a dispersão desses alvarás entre os bairros da cidade e entre ramos de atividade. Como métrica de dispersão (concentração) da distribuição de alvarás utilizamos a entropia de Shannon [Shannon 1948]. Dada uma distribuição de probabilidades $P = \{p_1, p_2, \dots, p_N\}$ com N elementos, onde $0 \leq p_i \leq 1$ e $\sum_i p_i = 1$, a entropia de Shannon H_S é definida como:

$$H_S = - \sum_{i=1}^N p_i \log_2 p_i. \quad (1)$$

No nosso caso, N pode identificar o número de bairros na cidade ou o número de ramos de atividade para se avaliar a distribuição de alvarás por bairro ou a distribuição de ramos de atividade por bairro, respectivamente. O valor p_i reflete a fração de alvarás em cada bairro ou a fração de alvarás por ramo de atividade. O valor de entropia mínimo $H_S^{\min} = 0$ indica concentração máxima. Por exemplo, o caso hipotético de todos os alvarás de uma cidade estarem registrados em um único bairro levaria à entropia mínima. Por outro lado, o valor de entropia máximo $H_S^{\max} = \log_2 N$ indica dispersão máxima, ou seja, uma distribuição uniforme dos alvarás entre os bairros da cidade levaria a um valor máximo de entropia. Para comparabilidade, consideramos a entropia normalizada $H_{\text{norm}} = \frac{H_S}{H_S^{\max}}$, onde $0 \leq H_{\text{norm}} \leq 1$, formando uma escala entre 0 e 1 indo da dispersão mínima até a dispersão máxima.

⁴<https://www.openstreetmap.org/> – Visitado em 15/05/2015.

⁵<http://www.qgis.org/en/site/> – Visitado em 15/05/2015.

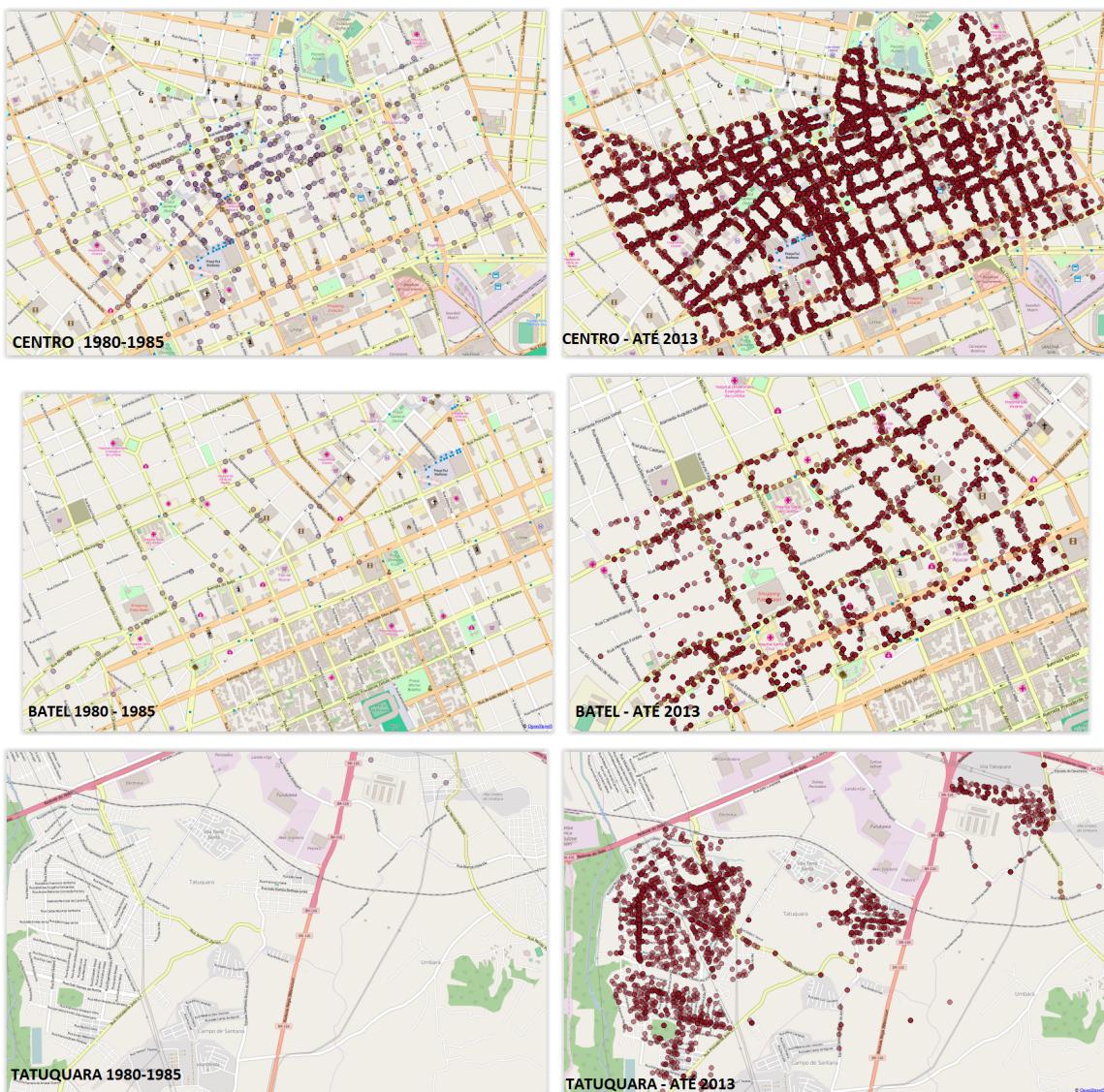


Figura 1. Alvarás concedidos no Centro, Batel e Tatuquara (1980-85 e até 2013).

De forma geral, a evolução temporal ao longo de três décadas da distribuição dos alvarás pelos 75 bairros considerados de Curitiba indica uma tendência ao longo dos anos de maior dispersão das atividades comerciais entre os bairros. O valor da entropia normalizada evolui de $H_{\text{norm}} = 0.80$ no quinquênio 1980-1984 para $H_{\text{norm}} = 0.90$ no período 2010-2013, indicando essa tendência de distribuição menos concentrada de alvarás concedidos entre os bairros da cidade ao longo do período analisado. Entretanto, essa maior distribuição de atividades pela cidade como um todo ocorre com maior concentração de algumas atividades em alguns bairros específicos, como discutiremos a seguir.

Nós também analisamos a evolução da atividade econômica em um período de mais de três décadas em três bairros de referência em Curitiba: Centro, Batel e Tatuquara (Figura 2). A Figura 2(a) mostra o crescimento do número de alvarás concedidos ao longo do período analisado. Como discutido anteriormente, ao longo do crescimento da atividade econômica houve também um processo de maior distribuição de atividades pelos bairros da cidade. Isso pode ser observado nos bairros do Batel e do Tatuquara que

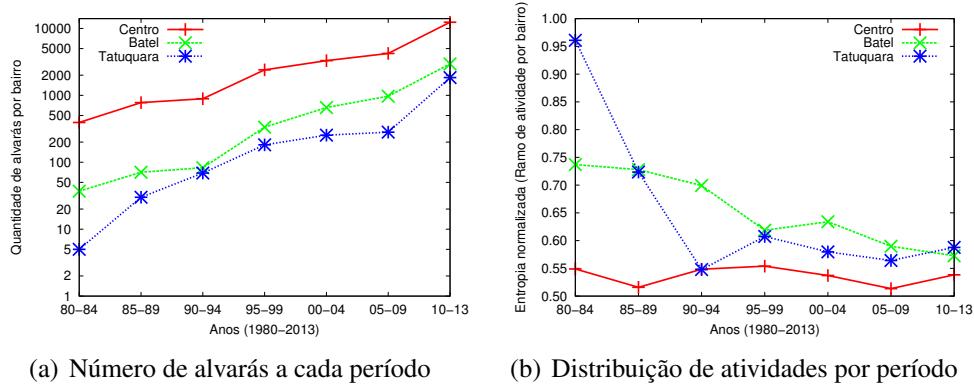


Figura 2. Evolução temporal da atividade econômica (Centro, Batel e Tatuquara).

tinham um número relativamente pequeno de alvarás no início do período de estudo, mas tiveram sua atividade econômica fortemente expandida ao longo do período analisado. Essa taxa de crescimento foi bem mais expressiva do que no centro da cidade.

A expansão econômica do Centro se deu mantendo o nível de concentração em alguns ramos de atividade, como mostra a evolução da entropia normalizada entre os ramos de atividade na Figura 2(b). Por outro lado, a forte expansão econômica em número de alvarás dos bairros Batel e Tatuquara se deu em meio a um processo de concentração (ou diminuição da maior dispersão — entropia normalizada) em algumas poucas atividades em cada bairro, cada um com seu perfil. No Batel, 74% das atividades em 2010-2013 se concentram em alvarás concedidos a escritórios, comércio varejista, serviços hospitalares e restaurantes. Em contraste, em 2010-2013, no Tatuquara, 77% das atividades se concentram em comércio varejista, escritórios, comércio atacadista e serviços de transporte.

4. Conclusão

O objetivo desta investigação inicial foi identificar o potencial de cenários e implicações, do ponto de vista de ciências de dados e geoprocessamento, utilizando três décadas de dados representando atividades econômicas de Curitiba. Estudos como este trazem uma análise preliminar não somente da expansão da atividade econômica da cidade ao longo das últimas décadas, mas também da dinâmica pela qual se deu essa expansão, destacando o seu potencial para o entendimento do processo de evolução econômica de uma cidade. Analisar essa dinâmica é o foco de nosso trabalho futuro, possivelmente integrando dados de redes sociais e fontes externas, além da avaliação de qualidade dos dados.

Agradecimentos

Os autores agradecem a RNP, CNPq, Prefeitura Municipal de Curitiba e IPPUC.

Referências

- Jat, M. K., Garg, P. K., and Khare, D. (2008). Modelling of urban growth using spatial analysis techniques: A case study of ajmer city (india). *IJRS.*, 29(2):543–567.
- Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656.
- Triantakonstantis, D. and Mountrakis, G. (2012). Urban growth prediction: A review of computational models and human perceptions. *J. of Geog. Inf. System*, 4(6):555–587.

aper:152862_1

DataUSP: Conjunto de serviços analíticos para apoio à tomada de decisões em uma instituição de ensino superior

Marino Hilário Catarino¹, Bruno Padilha¹, João Eduardo Ferreira¹

¹Instituto de Matemática e Estatística – Universidade de São Paulo (USP)

Rua do Matão, 1010 - CEP 05508-090 - São Paulo – SP – Brasil

marino@usp.br, padilha@ime.usp.br, jef@ime.usp.br

Abstract. This paper describes a Business Intelligence Application for decision-making in a higher education institution. The strategy adopted is described considering the business areas. The DataUSP main differential is in the best characterization of the indicator that is defined as quantifiable representation forms of services characteristics or processes, to monitor and improve results over time.

Resumo. Este artigo descreve uma aplicação de Business Intelligence para tomada de decisões em uma instituição de ensino superior. A estratégia adotada é descrita considerando as áreas de negócio. O maior diferencial do DataUSP está em caracterizar bem o indicador, que é definido como formas de representação quantificável de características de serviços ou processos, utilizado para acompanhar e melhorar os resultados ao longo do tempo.

1. Introdução

A tomada de decisões em uma instituição de ensino superior envolve não somente o corpo docente e discente vinculado à instituição, como também a responsabilidade social perante a sociedade em que se encontra inserida, o que amplifica sua importância [Calderón 2006]. Ao longo da evolução das instituições de ensino, diversos aplicativos operacionais foram desenvolvidas para atender tanto as necessidades acadêmicas quanto as administrativas, como recursos humanos e financeiro. Entretanto, percebe-se a necessidade de elaboração de um aplicativo voltado especificamente à gestão estratégica.

As necessidades operacionais em uma instituição de ensino superior acabam consumindo muito recurso para serem atendidas e, apesar de existir muitos estudos sobre a gestão em instituições [Godoy 2011], a oferta de um aplicativo dedicado à tomada de decisões nas diversas áreas de atuação não são encontradas. Um Planejamento Estratégico adequado é apontado em diversos trabalhos [Ferreira 2006] como uma solução. Apesar de já existirem no mercado algumas ferramentas de *Business Intelligence*, como o Tableau (www.tableau.com) e o SpagoBI (www.spagobi.org), a adequação ao ambiente acadêmico é esparsa e a esquematização em um aplicativo é custosa. Isso se deve às diversas características que compõe os indicadores de produtividade de uma instituição de ensino superior.

Tendo o objetivo de suprir esta necessidade, foi desenvolvido o aplicativo DataUSP, que contém diversas ferramentas analíticas para auxiliar na tomada de decisões em uma instituição de ensino superior, tendo como premissa atuar na

complexidade de informações de modo a sumarizar os dados mais relevantes a cada área atuação.

Além das informações existentes nas bases corporativas de uma instituição, o DataUSP dispõe de um conjunto de ferramentas para extração e análise dos dados dos programas de Pós-graduação da instituição que são enviados à Capes anualmente. Como resultado, fornece um relatório com histórico do extrato Qualis dos programas e o desempenho de cada docente, permitindo filtrar por períodos específicos.

Outro ponto importante para a análise de uma instituição é a quantidade de trabalhos publicados e o número de vezes que foi citado por outros autores, tanto interno quanto externos à instituição. As principais fontes que contabilizam o número de citações são o *Scopus*, *Web of Science* e o *Google Scholar*, sendo todas relevantes no meio acadêmico [Kulkarni 2009]. Através da quantidade de publicações e de suas respectivas citações define-se uma métrica quantitativa denominada índice-h [Hirsch 2005], que é uma proposta para quantificar a produtividade e o impacto de pesquisas individuais ou em grupos baseando-se nos artigos mais citados.

Todos os serviços do DataUSP apresentam inicialmente um panorama global da instituição, permitindo, através de uma navegação simples e intuitiva, iniciar com a visualização da informação sumarizada da instituição de ensino superior, percorrer informações específicas de uma unidade ou departamento e alcançar as informações de um docente. Essa estratégia de navegação permite percorrer da maior granularidade (instituição) para a menor (docente) de um modo ágil, e prático e intuitivo.

2. Metodologia Proposta

O projeto iniciou-se em 2012 com a modelagem dos dados para a criação de um *Data Warehouse*. Com o domínio dos dados bem definido, iniciou-se o processo de modelagem do sistema e a escolha das tecnologias que seriam utilizadas.

Uma mesma informação pode ser armazenada em um banco de dados com dois propósitos: ser operacional, atuando com inserção, deleção e atualização das informações, ou ser analítico, tendo como objetivo servir apenas para gerar relatórios e permitir uma análise dos dados.

Inicialmente foi realizado um espelhamento de todas as bases de dados corporativas que atendem cada negócio (graduação, financeiro, recursos humanos entre outros), deste modo mantendo a eficiência dos sistemas operacionais e permitindo que o sistema analítico atue de modo independente, não influenciando na performance do operacional. Estas novas bases espelhadas passaram por uma transformação nos dados que resultou no *Data Warehouse*, no qual são realizadas todas as atividades analíticas (Figura 1).

Como metodologia, foi adotada a orientação a serviços, sendo o aplicativo dividido em dois grandes componentes: servidor de recursos e interface web para a exibição dos dados. Para facilitar a escalabilidade, portabilidade e a manutenção do sistema, o DataUSP foi implementado com a linguagem de programação Java, seguindo as premissas do REST 2.4 [Mumbaikar 2013] por meio do framework Jersey. O formato *JavaScript Object Notation - JSON* foi adotado como padrão para a troca de mensagens. Com isso obteve-se uma formatação leve para troca de dados baseada em um subconjunto da linguagem de programação JavaScript.

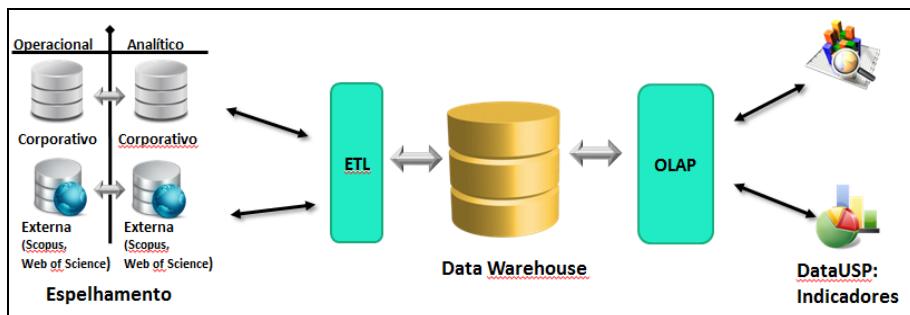


Figura 1. Estrutura do DataUSP

Quanto à visualização dos dados na interface web, foram definidos os seguintes requisitos: exibição de gráficos interativos em três dimensões e renderizados em tempo real; manipulação dos dados com Jquery (<http://jquery.com/>) e acesso ao servidor por meio de requisições assíncronas AJAX. A biblioteca *Fusion Charts* (www.fusioncharts.com) foi adotada por atender aos requisitos de integração com JSON e atuar com JavaScript, além de sua diversidade de modelos de gráficos (Figura 2).

Uma vez determinadas as tecnologias a serem utilizadas, iniciou-se o processo de construção de um arcabouço para execução das funcionalidades de *Web Services* do framework Jersey. O servidor Apache Tomcat, um container Web para executar aplicações que utilizam tecnologias Servlets e JSPs, foi adotado para instanciar as classes Java de acordo com as requisições web.

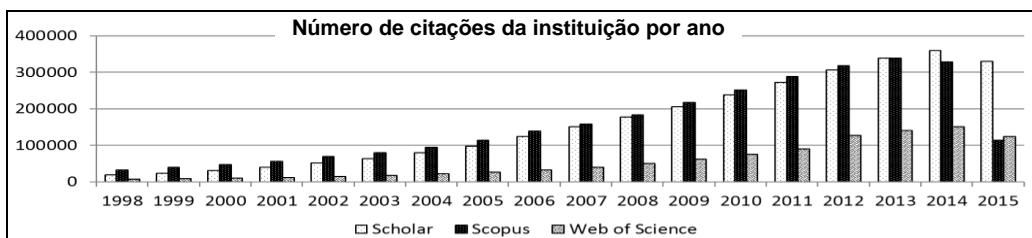


Figura 2. Gráfico das informações com opção de exportação em planilha CSV

Para receber o retorno do resultado de uma consulta, uma classe Java utiliza o *Data Access Object* – DAO. Este fornece uma API genérica para acessar dados em diferentes tipos de bancos de dados, executando uma consulta SQL ou uma chamada em uma Procedure e devolvendo o resultado em uma lista de objetos Java conforme cada consulta.

Por fim temos a classe de negócios, que recebe as requisições e, além de instanciar o DAO, também trata dos dados antes de enviar a resposta para seus métodos. Cada recurso de relatório do DataUSP possui esta estrutura, agrupados nas classes de negócio de acordo com sua funcionalidade. As classes de negócio correspondem às áreas de negócio da Universidade, que são: Ensino, Pesquisa e Extensão.

Quanto à visualização, os relatórios exibem os dados na forma de um gráfico interativo e também em uma tabela, que pode ser exportada como uma planilha no formato CSV, para permitir o manuseio dos dados conforme cada necessidade.

Apesar de cada área de negócio dispor de um conjunto próprio de indicadores, a estratégia de identidade visual adotada permite alternar entre as áreas seguindo um mesmo padrão.

3. Considerações Finais

Para se compreender a situação atual de uma instituição de ensino, é fundamental realizar uma análise abrangente e precisa dos dados acumulados ao longo dos anos. É nesse contexto que surgiu o DataUSP como conjunto de serviços analíticos da pós-graduação da Universidade de São Paulo para auxílio na tomada de decisão.

Os dados fornecidos pelo DataUSP são essenciais para se conhecer a evolução das diversas áreas de negócio, assim como para detectar as tendências e manter o padrão de excelência dos cursos e programas. Obter e tratar estes dados em tempo real é uma solicitação indispensável para maior agilidade na tomada de decisões e entendimento de anomalias que demandem ações estratégicas. A utilização de uma arquitetura orientada a serviços provou-se bastante adequada para o ambiente USP, garantindo alta escalabilidade de recursos computacionais e para a implementação de novas funcionalidades, facilitando a comunicação com outros futuros Web Services, e ainda, definindo um novo paradigma para se construir sistemas administrativos dentro da universidade.

Agradecimentos

Agradecemos a Fundação de Pesquisa de São Paulo pelo apoio financeiro (FAPESP concessão #2015/01587-0).

Referências

- Calderón, A. I. (2006) "Responsabilidade Social Universitária: Contribuições para o Fortalecimento do Debate no Brasil", Revista da Associação Brasileira de Mantenedoras de Ensino Superior, Brasília, v.24, n. 36, p. 7-22, jun. 2006.
- Ferreira, H. C. C., Ueno, E. M., Kovaleski, J. L. and Francisco, A. C. (2006), "Planejamento Estratégico, Ferramenta Indispensável para Gestão de Instituições de Ensino Superior IES Privadas", Em: III SEGeT – Simpósio de Excelência em Gestão e Tecnologia, Resende, 2006.
- Godoy, V. A. and Machado, M. (2011) "Planejamento Estratégico na Gestão Educacional: Uma Ferramenta Importante no Processo Decisório da Instituição de Ensino Superior", Revista Científica Intraciência, Ano 3, nº 3, p.32-85, Dez 2011.
- Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. Proceedings of the National Academy of Sciences, USA, 102(46), 16569–16572.
- Kulkarni, A. V. et al. (2009) "Comparisons of Citations in Web of Science, Scopus, and Google Scholar for Articles Published in General Medical Journals", JAMA - The Journal of the American Mecial Association, vol 302, n 10, Setembro de 2009.
- Mumbaikar, S., Padiya, P. (2013) "Web Services Based On SOAP and REST Principles", International Journal of Scientific and Research Publications, Volume 3, Issue 5, May 2013.
- Valcarenghi, E. V., Müller, I. R. F., Teza, P., Dandolini, G. A. and Souza, J. A. (2012) "Sistemas De Informação Como Ferramenta No Processo De Tomada De Decisão: O Caso Do Hu-Ufsc", Em: XI Congresso Brasileiro de Gestão do Conhecimento, KM Brasil 2012, São Paulo, 2012.

aper:152849_1

DeDup¹: um Aplicativo para Deduplicação de Contatos em Dispositivos Android

Rafael F. Pinheiro, Rafael Machado, Eliza A. Nunes, Eduardo N. Borges

Centro de Ciências Computacionais – Universidade Federal do Rio Grande (FURG)
Av. Itália, km 8, Campus Carreiros, Rio Grande – RS

{rafaelpinheiro, rafaelmachado, elizanunes, eduardoborges} @furg.br

Abstract. This paper presents an application for contact deduplication on Android devices called DeDup. This tool identifies duplicate contacts collected from different sources, such as e-mail accounts and social networks. Using multiple similarity functions, stored records are reorganized in groups of contacts representing the same person or organization.

Resumo. Este artigo apresenta um aplicativo para deduplicação de contatos em dispositivos Android denominado DeDup. Esta ferramenta identifica contatos duplicados provenientes de diferentes fontes, tais como contas de e-mail e redes sociais. Utilizando múltiplas funções de similaridade, os registros armazenados são reorganizados em grupos de contatos que representam a mesma pessoa ou organização.

1. Introdução

Com a explosão do número de aplicações Web disponíveis, os usuários tendem a acumular diversas contas em diferentes serviços como *e-mail*, redes sociais, *streams* de música e vídeo, lojas virtuais, entre outros. Gerenciar informações provenientes de múltiplos serviços a partir de um dispositivo móvel é uma tarefa complexa para o usuário. Alguns serviços básicos podem ser prejudicados pela redundância da informação coletada automaticamente por diferentes aplicações. Por exemplo, navegar na lista de contatos com tantas informações repetidas e muitas vezes incompletas reduz consideravelmente a produtividade que um *smartphone* pode oferecer.

A Figura 1 apresenta uma porção de uma base de contatos real composta por dez registros. Cada registro foi obtido de uma ou mais fontes de dados distintas representadas pelos ícones. Algumas informações já estão combinadas de duas ou mais fontes de dados, como é o caso do registro 3. Entretanto, os registros 4, 5, 6 e 8 representam a mesma pessoa e poderiam ser integrados ao registro 3 (grupo D). Idealmente, o resultado da deduplicação da lista de contatos são os grupos A, B, C e D.

O presente trabalho apresenta um aplicativo para dispositivos móveis Android [Ableson 2012], intitulado DeDup, capaz de identificar contatos duplicados provenientes de diferentes fontes de dados. A concepção e arquitetura do aplicativo foram publicadas anteriormente [Pinheiro et. al 2014]. Este trabalho apresenta um avanço significativo no método de deduplicação [Borges et al. 2011], pois são usadas funções de similaridade no lugar de comparações por igualdade. Além disso, o aplicativo proposto é comparado com outros sistemas disponíveis para Android.

¹ Demo disponível em <http://www.ginfo.c3.furg.br/>

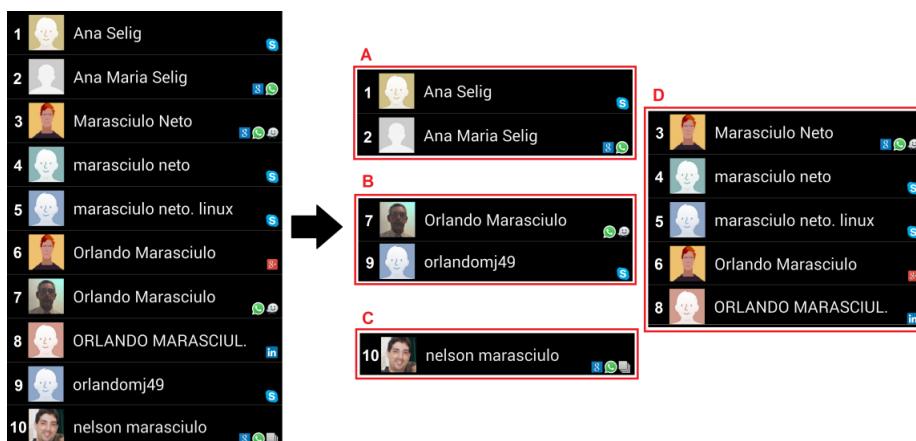


Figura 1. Exemplo de lista de contatos incluindo registros duplicados e o resultado esperado da deduplicação: grupos A, B, C e D.

2. Trabalhos Relacionados

O aplicativo Limpador de Contatos [Silva 2012] remove contatos duplicados da agenda comparando apenas números de telefone. A interface gráfica possui um único botão que quando acionado remove as duplicatas sem qualquer interação do usuário. Não é possível ver os contatos detectados como réplicas e tampouco restaurar a agenda original. Já Duplicate Contacts [Accaci 2015] permite visualizar aqueles com mesmo número de telefone e selecionar os registros a serem excluídos. Também é possível configurar um backup com a agenda no estado anterior às modificações. Duplicate Contacts Delete [Dabhi 2015] tem as mesmas funcionalidades dos anteriores, mas utiliza, além dos números de telefone, o nome do contato para identificar duplicatas. Contact Merger [ORGware Technologies 2015] integra todos os números de telefone de registros com o mesmo nome, mas mantém apenas um dos nomes do contato. Duplicate Contacts Manager [Sunil 2014] se destaca porque também utiliza o e-mail na deduplicação. A integração está disponível apenas na versão paga e não foi testada.

O diferencial do trabalho proposto neste artigo é o uso de similaridade textual para comparação dos nomes e e-mails, permitindo que muitos dos casos apresentados no exemplo motivacional da Figura 1 sejam identificados como o mesmo contato.

3. DeDup

O aplicativo desenvolvido coleta os contatos do dispositivo no qual está instalado provenientes da memória interna, cartão SIM e de contas vinculadas a outros aplicativos como mensageiros instantâneos e redes sociais. Para cada contato importado, são armazenados registros que contém campos que representem nome, telefone e e-mail. Os nomes são pré-processados removendo-se acentuação, caixa alta e caracteres diferentes de letras ou números. É armazenado em um novo campo o *login* do e-mail (sem o domínio). Por fim, são mantidos apenas os 10 algarismos finais do número de telefone.

Os registros são combinados em pares. Aqueles que compartilham pelo menos um número de telefone ou endereço de e-mail (casamento por igualdade) são identificados como duplicados. Sobre os demais registros são aplicadas as seguintes funções de similaridade [Cohen 2003] sobre seus campos: Levenshtein (logins), Jaccard (nomes), JaroWinkler (nomes) e Monge-Elkan (nomes).

Os escores de similaridade são combinados por uma média ponderada. Se o valor resultante é maior que um determinado limiar de similaridade, os registros são considerados equivalentes, ou seja, representam contatos duplicados. Os pesos e o limiar são definidos como parâmetros de configuração. Para agrupar os pares identificados como duplicados podem ser utilizados dois algoritmos [Kowalski 1997]: *single link* – cada registro é similar a pelo menos um registro do mesmo grupo; *click* – todos os registros de um grupo são similares entre si.

O DeDup foi implementado na linguagem Java, utilizando o kit de desenvolvimento de *software* (SDK) do Android. A Figura 2 apresenta a interface do DeDup. São exibidas da esquerda para a direita: a tela inicial; o menu de funções; a lista de contatos do dispositivo; e os pares identificados como duplicados junto da média ponderada dos escores retornados pelas funções de similaridade. O resultado do agrupamento de registros equivalentes pode ser visualizado nas telas correspondentes às opções do menu *Click* e *SingleLink* (não exibidas na figura por restrições de espaço).

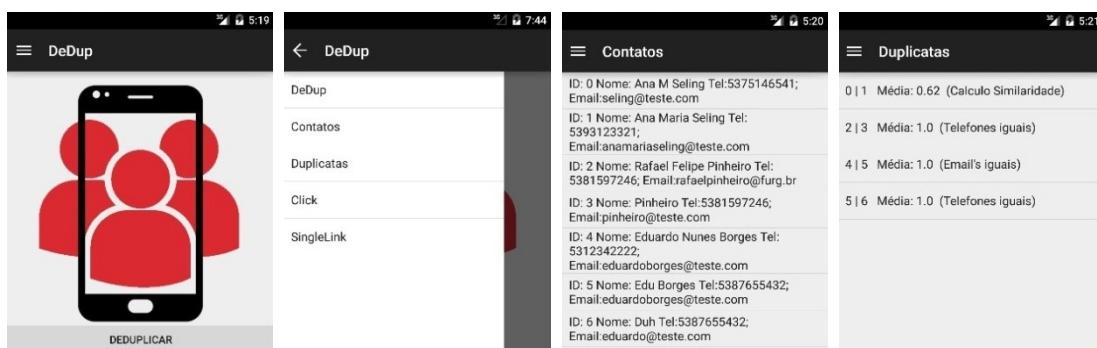


Figura 2. Interfaces do DeDup.

4. Experimentos

Os contatos apresentados na Figura 1 foram carregados no simulador BlueStacks², onde foram instalados, além do DeDup, todos os aplicativos relacionados na Seção 2. Limpador de Contatos e Duplicate Contacts não detectaram nenhum contato duplicado porque todos os registros têm números de telefone distintos. Duplicate Contacts Manager identificou apenas o grupo {6,7} porque é sensível a maiúsculas e minúsculas. Duplicate Contacts Delete e Contact Merger detectaram os grupos {3,4} e {6,7,8} como contatos duplicados. Além de agrupar o registro 7 incorretamente, deixou de agrupar todas as possíveis representações de Orlando Marasciulo Neto (grupo D). DeDup conseguiu agrupar corretamente todos os registros conforme o resultado apresentado na Figura 1 (grupos A, B, C e D), o que mostra sua real vantagem em utilizar funções de similaridade textual na comparação dos nomes próprios.

Foram realizados outros experimentos utilizando uma base de dados real com aproximadamente 2000 contatos importados de múltiplas fontes de dados: memória interna, cartão SIM, Skype, Facebook, LinkedIn, GMail e Google+. DeDup foi capaz de identificar corretamente até 76% dos pares similares duplicados, além de todos os pares com números de telefone ou e-mail iguais. Os detalhes [Machado 2015] foram omitidos neste artigo por restrições de espaço.

² <http://www.bluestacks.com>

5. Considerações finais

Este trabalho apresentou um aplicativo para deduplicação de contatos que facilita o processo de integração e reduz consideravelmente o tempo em que um usuário levaria para associar manualmente contatos de diversas contas. Como trabalhos futuros destaca-se a criação de um algoritmo mais complexo de detecção de duplicatas que utilize técnicas de aprendizagem de máquina. Estas técnicas devem aprender com os erros e acertos dos processos de deduplicação de cada usuário de forma a aperfeiçoar o processo para os demais, configurando automaticamente os pesos e limiar de similaridade adotados como padrão. DeDup ainda será reimplementado como um serviço para que a cada inserção ou exclusão de um contato, a deduplicação seja feita de forma incremental e bastante eficiente. A interface gráfica servirá apenas para configuração de parâmetros e interação com o algoritmo de integração, onde o usuário poderá escolher entre duas ou mais representações do nome de um contato duplicado.

Agradecimentos

Este trabalho está sendo parcialmente financiado pelos Programas Institucionais de Bolsas de Iniciação Científica, Tecnológica e de Inovação PROBIC/FAPERGS, PIBIC-PIBITI/CNPq e PDE/FURG.

Referências

- Ableson, W. F. *Android em ação*. Rio de Janeiro: Elsevier, 2012.
- Accaci, Alex (2015). Duplicate Contacts. Disponível em <http://play.google.com/store/apps/details?id=com.accaci>. Acesso: julho de 2015.
- Borges, E. N., Becker, K., Heuser, C. A. & Galante, R. (2011). A classification-based approach for bibliographic metadata deduplication. In: Proceedings of the IADIS International Conference WWW/Internet, p. 221-228, Rio de Janeiro.
- Cohen, W., Ravikumar, P., & Fienberg, S. (2003). A comparison of string metrics for matching names and records. In: KDD Workshop on Data Cleaning and Object Consolidation, vol. 3, p. 73-78.
- Dabhi, Pradip (2015). Duplicate Contacts Delete. Disponível em <http://play.google.com/store/apps/details?id=com.don.contactdelete>. Acesso: julho de 2015.
- Kowalski, G. (1997). *Information Retrieval Systems: Theory and Implementation*. Kluwer Academic Publishers, Norwell, MA, USA.
- Machado, R. F. (2015). Identificação de contatos duplicados utilizando similaridade textual e aprendizagem de máquina. Monografia de Graduação (Engenharia de Computação), Centro de Ciências Computacionais, FURG, Rio Grande.
- ORGware Technologies (2015). Contact Merger. Disponível em <http://play.google.com/store/apps/details?id=com.orgware.contactsmerge>. Acesso: julho de 2015.
- Pinheiro, R., Lindenau, G., Zimmermann, A., Borges, E. N. (2014). Um aplicativo para integração de contatos em dispositivos Android. In: Anais do Congresso Regional de Iniciação Científica e Tecnológica em Engenharia, p. 1-4. Alegrete.
- Silva, Alan Martins (2012). Limpador de Contatos. Disponível em <http://play.google.com/store/apps/details?id=br.com.contacts.cleaner.by.alan>. Acesso: julho de 2015.
- Sunil D M (2014). Duplicate Contacts Manager. Disponível em <http://play.google.com/store/apps/details?id=com.makelifesimple.duplicatedetector>. Acesso: julho de 2015.

RelationalToGraph: Migração Automática de Modelos Relacionais para Modelos Orientados a Grafos

Gabriel Zessin, Edson Oliveira Jr

Departamento de Informática (DIN) – Universidade Estadual de Maringá (UEM)
87020-900 – Maringá – PR – Brasil

gabrielzessin@gmail.com, edson@din.uem.br

Abstract. This paper presents an ongoing development of an application to automatically migrate from an existing relational data model in a database to a graph-oriented model. The application is responsible for communicating with the existing database, collecting metadata and generating a graph-oriented model that accurately represents the original one.

Resumo. Este artigo apresenta o desenvolvimento em andamento de uma aplicação com o propósito de realizar automaticamente a migração de um modelo de dados relacional já existente em uma base de dados para um modelo orientado a grafos. A aplicação é responsável por se comunicar com a base existente, coletar metadados e gerar um modelo orientado a grafos que representa fielmente o original.

1. Introdução

Com o crescimento de aplicações que geram dados não estruturados ou semiestruturados em escala muito grande, o uso de bancos de dados não relacionais tem sido adotado para suprir as necessidades que tais aplicações demandam. Esses bancos de dados começaram a ter seu espaço no mercado, pois atendem justamente a algumas deficiências que os bancos relacionais possuem: a escalabilidade e o desempenho em aplicações que trabalham com informações não estruturadas e que crescem de forma muito rápida (na ordem de Petabytes de informações) [Kaur e Rani 2013].

Os bancos de dados não relacionais (comumente chamados de NoSQL) são divididos em diferentes tipos de acordo com a modelagem dos dados, sendo eles: chave-valor; orientado a documentos; orientado a colunas e orientado a grafos [Kaur e Rani 2013]. Cada um desses tipos tem suas particularidades e é aplicado de acordo com a necessidade.

Apesar dos bancos de dados NoSQL estarem se tornando cada vez mais populares, ainda não existem muitas ferramentas que auxiliem no seu gerenciamento ou na migração partindo de um modelo relacional. Para os bancos de dados orientados a grafos, algumas ferramentas encontradas fazem o trabalho da migração dos dados de uma base relacional para uma orientada a grafos, sendo elas: uma ferramenta própria no Neo4j, que é um dos SGBDs orientados a grafos mais conhecidos; e a ferramenta R2G [Maccioni et al 2013], que faz a tradução de consultas SQL além da migração dos dados.

No entanto, o objetivo da aplicação *RelationalToGraph*, que vem sendo desenvolvida em nosso grupo de pesquisa, não é realizar a migração dos dados de fato,

mas sim do modelo que os definem. Assim, tal modelo orientado a grafos pode ser melhor analisado antes da migração dos dados ser realizada. Logo, a aplicação independe de uma nova base de dados para realizar a migração, pois ela é responsável por gerar um grafo a partir do modelo que define os dados existentes em uma base de dados relacional – aspecto que difere a aplicação *RelationalToGraph* das demais ferramentas encontradas.

Nas próximas seções, o modelo relacional adotado para ilustrar a aplicação desenvolvida é apresentado. Feito isso, apresenta-se como a migração para um modelo orientado a grafos é realizada. Por fim, são apresentados e analisados os resultados obtidos ao executar a aplicação em uma base de dados relacional que contém o mesmo modelo utilizado como exemplo nos passos anteriores.

2. O Modelo Relacional de Exemplo

Para ilustrar a execução da aplicação desenvolvida é adotado o modelo de base de dados relacional apresentado por Elmasri e Navathe (2011) (Figura 1).

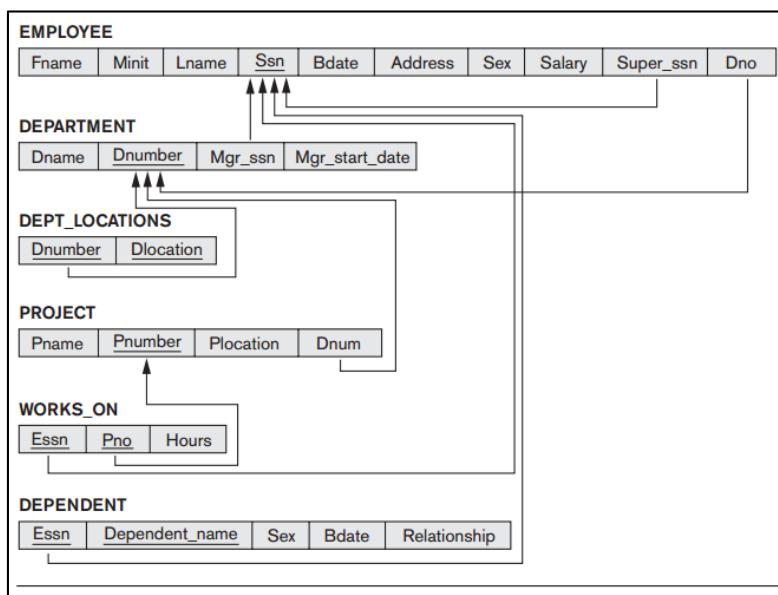


Figura 1. Modelagem relacional adotada [Elmasri e Navathe, 2011, p. 91]

3. A Migração do Modelo Relacional para Orientado a Grafos

Com base no modelo apresentado, é possível seguir alguns passos para gerar um grafo que o represente [Maccioni et al 2013]. O primeiro ponto a ser destacado é que, para a abordagem de problemas na forma de um grafo, o que é modelado de fato são os dados que existem na base, não o esquema que define esses dados. Isso deve-se ao fato de que os bancos de dados não relacionais são capazes de armazenar os dados em uma forma mais natural, sem uma estrutura rígida que deve ser seguida [Kaur e Rani 2013].

Com isso, um vértice com nome “Employee” no grafo, por exemplo, não representa a entidade “Employee” do modelo relacional, mas sim um registro do tipo “Employee”. Ou seja, embora na Figura 2, que representa o grafo que foi criado, só exista um nó chamado “Employee”, nada impede que mais nós como esse sejam adicionados ao grafo. Porém, a estratégia adotada é a de criar somente um vértice para representar cada entidade do modelo relacional.

Já para representar os atributos das entidades do modelo relacional, são criadas propriedades nos vértices ou arestas do grafo.

As chaves estrangeiras do modelo relacional passam a ser representadas por arestas no grafo. A relação entre as entidades é representada por arestas. Outro ponto importante é que as entidades que servem exclusivamente para tratar de relacionamentos do tipo “muitos para muitos” podem ser removidas do modelo, bastando adicionar arestas que representem essas relações no grafo, lembrando que os atributos dessas entidades passam a ser propriedades das arestas que as representam. Entendendo tais princípios, é possível analisar a Figura 2, que representa a modelagem orientada a grafos para o modelo de exemplo utilizado na Figura 1.

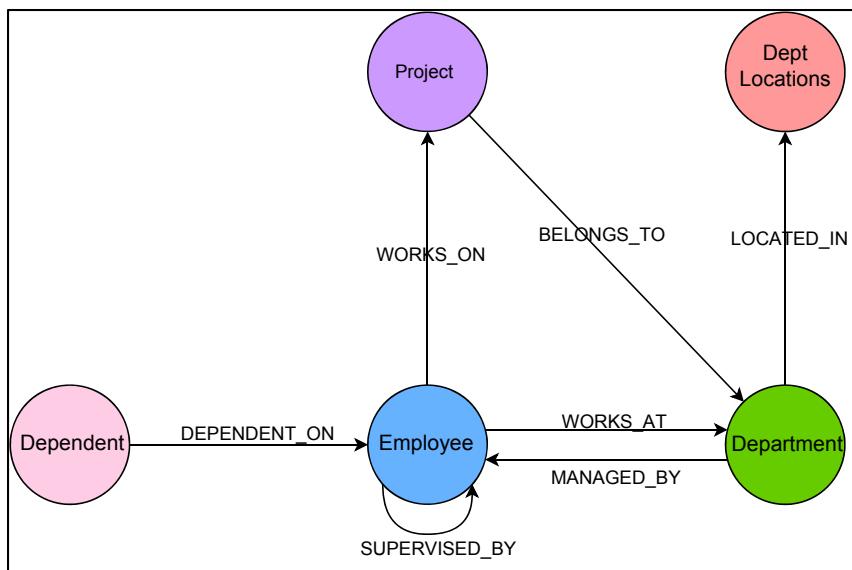


Figura 2. Grafo criado manualmente para o modelo relacional adotado

4. Resultados Obtidos com a Aplicação

A aplicação *RelationalToGraph* foi executada em uma base de dados MySQL, embora ela possa ser executada em outros SGBDs, como Oracle e PostgreSQL. Tal base possuía as entidades e relações do modelo de exemplo utilizado. Para executá-la, as informações de conexão e algumas outras necessárias para o seu funcionamento foram passadas à aplicação por meio de um arquivo de propriedades. A Figura 3 representa o grafo gerado automaticamente para o modelo considerado.

Comparando o grafo gerado pela aplicação com o grafo criado manualmente (Figura 2), percebe-se que o modelo de ambos é quase idêntico. A única coisa que os diferencia é a direção da aresta entre os vértices “Dept_locations” e “Department”. Por meio de testes empíricos realizados no SGBD orientado a grafos Neo4j, a direção da aresta não tem grande interferência no momento de realizar uma consulta. No entanto, para uma melhor análise do modelo orientado a grafos, a direção das arestas é muito importante – de fato, é uma dificuldade para a aplicação decidir qual a direção para a aresta que, semanticamente, melhor represente a relação do modelo original.

5. Considerações Finais

Os resultados obtidos com a aplicação foram satisfatórios, dado que os grafos gerados para os modelos nos quais a aplicação foi executada representaram fielmente os modelos originais. No entanto, alguns detalhes podem ser melhorados, tais como: a disposição e o nome das arestas geradas e a exibição das propriedades de cada vértice e aresta, visto que os atributos das entidades agora passaram a ser propriedades desses elementos; e a abordagem de outras técnicas de modelagem NoSQL, tal como a NoAM [Atzeni et al 2014] – uma modelagem abstrata para os quatro tipos de bancos de dados não relacionais.

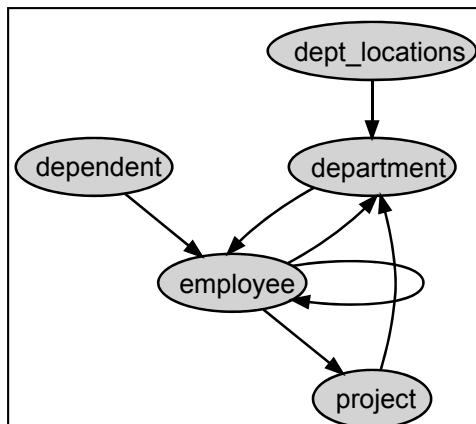


Figura 3. Grafo gerado automaticamente pela aplicação para o modelo relacional adotado

Por fim, vale ressaltar que a aplicação consegue se comunicar com vários SGBDs relacionais, sendo simples a adição de suporte a novos SGBDs. Além disso, ela é capaz de identificar entidades que servem para representar relacionamentos do tipo “muitos para muitos”. Nesse exemplo, analisando o modelo relacional da Figura 1, percebe-se que a entidade “Works_on” serve para fazer o relacionamento “muitos para muitos” entre as entidades “Employee” e “Project”. Tais entidades são criadas pois não existe outra forma mais correta de tratar esse tipo de relacionamento no modelo relacional. Já no modelo orientado a grafos, elas simplesmente deixam de existir, sendo substituídas por uma aresta entre essas duas entidades que se relacionam. Analisando o grafo gerado da Figura 3, é possível verificar que, de fato, isso ocorreu, pois uma aresta entre “Employee” e “Project” foi adicionada, sendo que não existe relação direta entre elas no modelo relacional.

Referências

- Atzeni, P.; Bugiotti, F.; Cabibbo, L.; Torlone, R. Database Design for NoSQL Systems. International Conference on Conceptual Modeling, Atlanta, United States, pages 223 - 231, oct. 2014.
- Elmasri, R.; Navathe, S. B. (2011), Fundamentals of Database Systems – 6a. Ed. Addison-Wesley.
- Kaur, K.; Rani, R. Modeling and Querying Data in NoSQL Databases. IEEE International Conference on Big Data, Silicon Valley, United States, pages 1-7, oct. 2013.
- Maccioni, A.; Torlone, R.; Virgilio, R. Converting Relational to Graph Databases. Grades '13 First International Workshop on Graph Data Management Experiences and Systems, New York, United States, article no. 1, pages 1-6, jun. 2013.

aper:152911_1

Uma proposta para apresentar a Computação/Banco de Dados no Ensino Médio para o Público Feminino

Juan J. Rodriguez¹, Nádia P. Kozievitch¹, Silvia A. Bim¹, Mariangela de O. G. Setti¹,
Maria C. F. P. Emer¹, Marília A. Amaral¹

¹Dep. de Informática, UTFPR, Curitiba, PR, Brasil

jrodriguezv10@gmail.com, nadiap@utfpr.edu.br,

{sabim, mari, mclaudia, mariliaa}@dainf.ct.utfpr.edu.br

Abstract. *The number of women in computer science is still low compared to the total number of professionals within this area. Several actions are being held in Brazil and other countries to reverse this scenario. In this paper, we present one of such actions in the context of a project, which aims to bring computer science to female high school students in Curitiba - Brazil.*

Resumo. *O número de mulheres na computação ainda é baixo se comparado ao número total de profissionais nesta área. Várias ações estão sendo realizadas no Brasil e em outros países para reverter este cenário. Este trabalho relata uma das atividades de um projeto que tem como objetivo apresentar as diferentes facetas da computação para estudantes femininas do ensino médio de uma escola pública em Curitiba - Brasil.*

1. Introdução

A constante integração da computação no cotidiano das pessoas e a diversidade de tópicos a serem estudados e aplicados não são suficientes para atrair talentos para a área de Computação. Alguns trabalhos [Beaubouef and McDowell 2008, Beaubouef and Mason 2005] têm analisado o declínio do número de alunos nos cursos de ciência da computação (CS) e áreas correlatas. Em particular, estudos mundiais [Beaubouef and Zhang 2011] indicam que o número de estudantes do sexo feminino é bem menor que o número de estudantes do sexo masculino.

No cenário brasileiro, a realidade também não é diferente, desde os anos 80 há uma expressiva diminuição da quantidade de mulheres que concluem os cursos da área de Computação no Brasil. Já foi constatado que alguns estados (como a Paraíba) têm uma representatividade ainda menor de mulheres em cursos de computação [Oliveira et al. 2014b]. Várias ações estão sendo realizadas no Brasil (como a análise de perfil [Oliveira et al. 2014a]) e em outros países na tentativa de despertar o interesse de jovens mulheres pelos cursos da área de Computação. A grande maioria destas iniciativas apresenta a Computação por meio de uma única perspectiva - a programação. Entretanto, algumas iniciativas desenvolvem atividades que exploram outras áreas da Computação como Interação Humano-Computador [Maciel et al. 2013] e Banco de Dados [Martinhago et al. 2014].

Diante deste cenário, este trabalho pretende divulgar a computação, descrevendo uma oficina realizada na UTFPR, que aborda o ensino de Banco de Dados às alunas do

ensino médio. O trabalho está organizado da seguinte maneira: a Seção 2 apresenta a metodologia e desenvolvimento da oficina, a Seção 3 apresenta os resultados, e finalmente, a Seção 4 apresenta a conclusão e os trabalhos futuros.

2. Metodologia e Desenvolvimento

A oficina foi realizada de 12 a 15 de agosto de 2014, com 16 alunas do Colégio Estadual Dr. Xavier da Silva¹. Além das alunas, uma das professoras da escola e três voluntários da graduação do Curso de Bacharelado em Sistemas da Informação da UTFPR (participantes do PET-CoCE² participaram na administração das aulas e revisão do material utilizado). Além disso, alunos do mestrado profissional da mesma instituição (PPGCA³) ajudaram na revisão do material de ensino.

A oficina teve o objetivo de instigar a curiosidade sobre temas da computação (em particular, Banco de Dados) nas alunas do ensino médio. A metodologia usada para este trabalho foi composta pelas seguintes etapas: o levantamento dos temas de banco de dados a serem abordados, a integração com alunos de graduação e pós-graduação na elaboração do material de ensino, a realização da oficina, e o incentivo ao estudo de tópicos avançados. A Figura 1 ilustra a integração dos grupos e os temas abordados na Oficina de Banco de Dados. De maneira resumida, o conhecimento de docentes, alunos de graduação, de pós-graduação, e de aplicações externas foram contextualizados para o ensino médio, em temas como Bibliotecas Digitais, Educação, Redes Sociais, entre outros.

Na etapa inicial foi definida a abordagem dos seguintes tópicos: (i) **Conceitos Básicos de Banco de Dados:** BD, SGBD, Modelo Relacional e tipos de dados; (ii) **Motivação:** nesta etapa foram ilustradas diferentes aplicações dentro da área de Banco de Dados, como as Bibliotecas Digitais (Biblioteca Digital da Universidade de Kabul⁴, Biblioteca Digital de Livros Raros da Armênia⁵, Biblioteca Digital de Médicos e assuntos ligados a medicina⁶, entre outros), redes sociais (como o grupo brasileiro *Women in Databases*⁷), aplicações de geoprocessamento, etc; (iii) **Tipos de Bancos de Dados:** Hierárquico, Rede, O.O., Relacional, etc.; (iv) **Ideia básica de otimização:** o que é um índice, quais seus tipos, exemplos; e (v) **Exemplo simples de uso de Bancos de Dados** (como o banco de dados de alunos, Mapeamento de rios, represas, áreas indígenas e nascentes da COPEL⁸) e iteração com projetos (como a página do próprio projeto Emíli@as⁹).

Na segunda etapa foi realizada a integração dos alunos de graduação e pós-

¹<http://www.ctaxaviersilva.seed.pr.gov.br/modules/noticias/> Último acesso em 10/06/2015.

²<http://www.dainf.ct.utfpr.edu.br/petcoce/> Último acesso em 10/06/2015

³<http://ppgca.dainf.ct.utfpr.edu.br/doku.php> Último acesso em 10/06/2015

⁴<http://puka.cs.waikato.ac.nz/cgi-bin/library?a=p&p=about&c=acku> Último acesso em 10/06/2015.

⁵<http://greenstone.flib.sci.am/gsdl/cgi-bin/library.cgi> Último acesso em 10/06/2015.

⁶<http://library.medicine.yale.edu/find/digital> Último acesso em 10/06/2015.

⁷<https://www.facebook.com/groups/womenindb/>

⁸<http://www.copel.com/sig-sam/sig-sam.jsp> Último acesso em 10/06/2015.

⁹<http://emili@as.dainf.ct.utfpr.edu.br/> Último acesso em 10/06/2015.

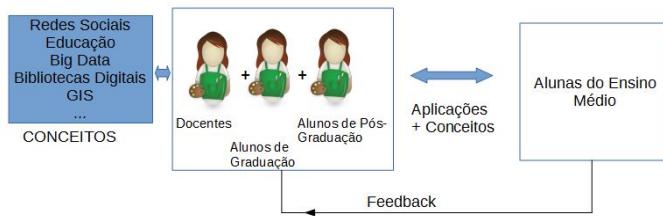


Figure 1. A integração de grupos e temas na Oficina de Banco de Dados.

graduação na elaboração do material de ensino. O material (conjunto de slides inicialmente feito pelos alunos de graduação e a professora da oficina) foi debatido e melhorado pelos alunos de pós-graduação. O objetivo foi aproveitar o conhecimento de mercado dos alunos de pós-graduação para melhorar a linguagem, os exemplos, os exercícios, a conexão com temas externos, junto com a abordagem da apresentação. Dentre os exemplos de exercícios, pode ser citada a criação de tabelas em excel, com atributos e tipos diferenciados, para incentivar a definição de colunas, tuplas, ordenação, índices, entre outros.

As aulas das oficinas foram realizadas em um laboratório com o maquinário Macintosh, utilizando slides (para partes teóricas), e exercícios (para a compreensão do impacto prático). O material utilizava como exemplo aplicações com temas atuais (como redes sociais, YouTube, integração de imagens, etc), temas de pesquisa (bibliotecas digitais, geoprocessamento, entre outros), e materiais que possibilitassem continuar o aprendizado (em fontes externas, como banco de dados e aplicações para crianças^{10 11}, tutoriais de SQL¹², entre outros).

3. Resultados

Dentre os resultados, podem ser citadas as considerações dos participantes (alunos, voluntários e professores) sobre o evento e as dificuldades enfrentadas. Sobre a experiência de trabalhar na oficina de Banco de Dados, um dos voluntários comentou que *"Foi uma ótima experiência e recomendaria para outros alunos, pois serviu de base para solidificar meus conhecimentos da área, e além disso, me ajudou a mapear uma perspectiva para outras oficinas que desenvolvi e apliquei dentro da Universidade. Com a oficina foi possível desenvolver uma visão crítica sobre o que é computação para pessoas que apenas a utilizam como usuários comuns. Isso me ajudou a perceber uma nova área de pesquisa baseada na desmistificação da computação"*.

A professora de ensino médio que acompanhou as alunas (e que já possuía o curso técnico em informática) afirmou que: *"o curso foi muito claro, e de maneira fácil, proporcionou o entendimento sobre banco de dados, bem como a sua importância, dentro de um programa ou até mesmo dentro de uma empresa"*.

A participação recorrente das alunas e a ausência de evasão em todas as oficinas seguintes reflete que a oficina de Banco de Dados despertou o interesse pela computação.

¹⁰<http://www.purplemash.com/#/home> Último acesso em 10/06/2015

¹¹http://www.teachingideas.co.uk/ict/contents_spreadsheetsdatabases.htm Último acesso em 10/06/2015

¹²<http://www.sqlcourse.com/intro.html> Último acesso em 10/06/2015

Um ano após a atividade um questionário foi enviado às participantes, perguntando se (i) a oficina havia incentivado a gostar de computação, se (ii) a aluna faria mais oficinas do mesmo tema, e (iii) do que a aluna havia gostado mais. Todas as participantes relataram que se lembravam do conteúdo, e 2/3 das respostas indicaram que gostariam de fazer outras oficinas no mesmo tema. Uma das participantes comentou que "*Esta foi a oficina que mais gostei, teve ótimas explicações e o assunto me envolveu, me deixando com curiosidade sobre cursar o tema*". Além disso, foi realizado um painel, no qual as alunas definiam com uma palavra o que lembravam da oficina. Os comentários, em geral, ressaltaram aspectos positivos como "*interessante*", "*diferente*" e "*prático*".

Considerando as dificuldades e desafios enfrentados, é importante observar como resultados da oficina: (i) o tratamento de temas teóricos de Banco de Dados para instigar alunas do ensino médio; (ii) a integração de alunos de graduação e pós-graduação na problemática da percepção de leigos sobre a Computação; (iii) a integração de conteúdos dinâmicos da Web (Sites, Redes Sociais, etc.) para atrair a atenção das alunas; e (iv) a prática de exercícios baseados em aplicações atuais (Facebook, YouTube, etc.), ilustrando conceitos de banco de dados e computação.

4. Conclusão e Trabalhos Futuros

Várias ações estão sendo realizadas no Brasil e em outros países do mundo na tentativa de despertar o interesse de jovens mulheres pelos cursos da área de Computação. Em particular, este trabalho abordou a problemática por meio do estímulo da computação (em particular, Banco de Dados) em alunas do ensino médio. Com base na pesquisa obtida junto aos participantes, pode-se dizer que o objetivo de instigar a curiosidade pela Computação foi alcançado. Como trabalhos futuros, pretende-se realizar oficinas com níveis de dificuldades e mídias diferenciadas.

References

- Beaubouef, T. and Mason, J. (2005). Why the high attrition rate for computer science students: Some thoughts and observations. *ACM Special Interest Group on Computer Science Education Bulletin*, 37(2):103–106.
- Beaubouef, T. and McDowell, P. (2008). Computer science: Student myths and misconceptions. *J. Comput. Sci. Coll.*, 23(6):43–48.
- Beaubouef, T. and Zhang, W. (2011). Where are the women computer science students? *J. Comput. Sci. Coll.*, 26(4):14–20.
- Maciel, C., Bim, S. A., and Boscaroli, C. (2013). Hci with chocolate: Introducing hci concepts to brazilian girls in elementary school. In *CLIHC 2013 - Volume 8278*, pages 90–94, New York, NY, USA. Springer-Verlag New York, Inc.
- Martinhago, A., Smarzaro, R., Lima, I., and Guimarães, L. (2014). Computação desplugged no ensino de banco de dados na educação superior. In *XXII WEI*.
- Oliveira, A., Moro, M., and Prates, R. (2014a). Perfil feminino em computação: Análise inicial. In *XXII Workshop sobre Educação em Computação*, pages 1465–1474.
- Oliveira, M., Souza, A., Barbosa, A., and Barreiros, E. (2014b). Ensino de lógica de programação no ensino fundamental utilizando o scratch: um relato de experiência. In *XXII WEI*, pages 1525–1534.

xml2arff: Uma Ferramenta Automatizada de Extração de Dados em Arquivos XML para *Data Science* com Weka e R

Gláucio R. Vivian¹, Cristiano R. Cervi¹

¹Instituto de Ciências Exatas e Geociências (ICEG)
Universidade de Passo Fundo (UPF) – Passo Fundo – RS – Brazil

{149293, cervi}@upf.br

Abstract. This short paper reports the development of a tool to assist researchers at *Science Data* in extraction of data from XML files to software Weka and R. We use the query language XQuery to recover the information. The proposed tool performs an automated process of data analysis. Finally there is the possibility of exporting data in native format for softwares such Weka and R. This automation results in a significant reduction of time between data retrieval and processing when compared to manual processing.

Resumo. Este artigo relata o desenvolvimento de uma ferramenta para auxiliar os pesquisadores de *Data Science* na extração de dados em arquivos XML para os softwares Weka e R. Utilizamos a linguagem de consulta XQuery para recuperarmos as informações. A ferramenta proposta executa um processo automatizado de análise dos dados. Finalmente existe a possibilidade de exportação dos dados em formatos nativos para softwares como Weka e R. Essa automatização resulta em uma redução significativa de tempo entre a recuperação dos dados e seu processamento quando comparado ao processamento manual.

1. Introdução

O armazenamento e intercâmbio de dados em arquivos XML(*Extensive Markup Language*) está cada vez mais frequente no cotidiano dos pesquisadores. Esse formato de arquivo é definido e recomendado pela W3C(*World Wide Web Consortium*). Existem inúmeras tecnologias projetadas para interagir com o formato XML, cada uma com um propósito específico. Dentre elas a linguagem XQuery¹ permite a realização de consultas diretamente em arquivos XML. Seu poder de expressão é equiparado à linguagem SQL para banco de dados relacionais.

Em *Data Science* existem diversos softwares para análise dos dados. Para estudos com características estatísticas utiliza-se a linguagem R[Jaffar et al. 1992][Gentleman et al. 2009]. No caso de mineração de dados e aprendizagem de máquina, a ferramenta Weka[Hall et al. 2009] é bastante utilizada por apresentar inúmeras opções de algoritmos. Essas ferramentas se caracterizam por necessitarem da entrada dos dados em um formato predefinido. Existem algumas rotinas de importação de dados, mas que necessitam da interação do usuário para execução e que muitas vezes pela sua característica repetitiva acaba tornando a tarefa onerosa.

No trabalho de [Hornik et al. 2009] espoem-se a necessidade de integração entre o software de estatística R com o Weka. Esta integração torna possível o aproveitamento de

¹<http://www.w3.org/TR/xquery/>

técnicas de aprendizado de máquina/descoberta de conhecimento. Os autores apresentam o pacote RWEKA[Hornik et al. 2007] para R. O mesmo possibilita a utilização dos algoritmos do Weka através das suas interfaces funcionais disponíveis. O RWEKA apresenta a desvantagem de acessar apenas um subconjunto dos recursos do Weka e necessita ser atualizado periodicamente conforme o Weka for evoluindo.

O objetivo deste trabalho é apresentar uma ferramenta que permite a extração automatizada de dados em arquivos XML para o formato nativo dos *softwares* Weka e R usados em análises de *Data Science*. Tal processo simplifica o intercambio de informações reduzindo significativamente o tempo despendido para tal tarefa.

Este artigo está organizado da seguinte forma: Seção 2 apresentação da ferramenta. Seção 3 apresenta os experimentos e resultados. Finalmente na seção 4 as considerações finais e sugestões de trabalhos futuros.

2. A ferramenta xml2arff

A ferramenta xml2arff(lê-se: XML to ARFF) permite a conversão dos dados em arquivos XML para o formato nativo das ferramentas de *Data Science*. A ferramenta utiliza a linguagem XQuery como padrão para consultas. As consultas realizadas em arquivos locais utilizam a biblioteca Saxon². Caso os arquivos estejam em outro repositório também é possível realizar a conexão cliente/servidor com o BaseX Server³ utilizando o protocolo TCP/IP. A consulta deve ser projetada para retornar dados de forma não hierárquica, ou seja, todos os atributos em uma linha. O resultado da consulta é processado pelo algoritmo de automatização. Esta etapa busca identificar as características dos atributos(tipo e frequência) para posteriormente gerar os metadados e formatar os atributos para o formato de destino(CSV ou ARFF). O resultado é apresentado ao usuário para avaliação, que pode livremente alterar a nomenclatura e tipo do atributo se assim o desejar. Na figura 1 pode-se visualizar todas as etapas de funcionamento da ferramenta proposta. Na figura 2 pode-se visualizar o diagrama de classes.

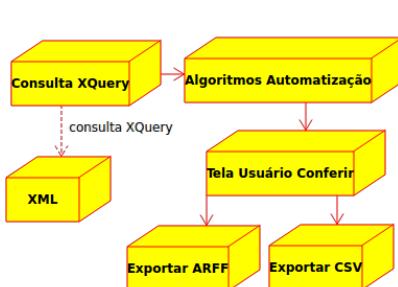


Figura 1. Etapas do processo

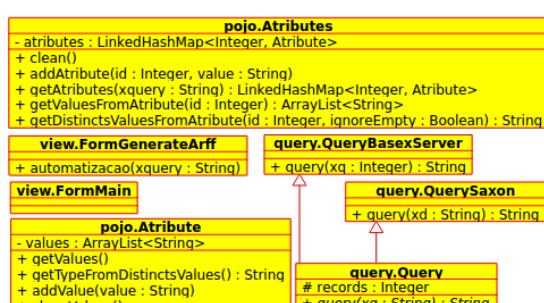


Figura 2. Diagrama de Classes

O formato de arquivo nativo do Weka é o ARFF(*Attribute Relation File Format*). Neste formato os atributos podem ser de três tipos. Os dados numéricos são representados como *numeric*. Caso o dado apresente alta frequência de repetição, ou seja, com poucos valores diferentes, o tipo deve ser *nominal*. Esta diferenciação em *nominal* permite ao Weka otimizar a utilização de recursos computacionais. Caso contrário, o dado será formatado como *string*. O formato também prevê a existência de valor vazio para o atributo.

²<http://saxon.sourceforge.net>

³<http://baseX.org>

Para mais detalhes sobre o formato ARFF vide [Holmes et al. 1994]. No caso do *software R*, o padrão para os arquivos é o CSV(*Comma Separated Value*). Neste formato os atributos são separados por uma vírgula. Geralmente a primeira linha do arquivo possui a nomenclatura dos atributos. O separador decimal deve ser o ponto. As *strings* devem ser encapadas com aspas duplas.

2.1. Algoritmo de Automatização

A etapa de automatização é realizada por um algoritmo específico. Ele realiza a análise dos dados para cada atributo da consulta. A partir disso, identificamos o tipo de dado e sua frequência. Essas informações serão úteis para gerar os metadados e formatar os valores de acordo com tipo de dados. A complexidade assintótica é de $O(\text{atributos} * \text{valores})$. O algoritmo foi implementado na linguagem Java seguindo a documentação UML da figura 2. A seguir pode-se visualizar o algoritmo em pseudocódigo.

Algoritmo 1 Algoritmo de Automatização

```

1: função AUTOMATIZACAO(xquery, maxDistinctValuesForNominal)
2:   atributos ← Atributes.getAtributes(xquery)
3:   para cada i from atributos.size() faça
4:     numero ← 0
5:     valoresUnicos.clean()
6:     para cada j from atributos.getValues().get(i).size() faça
7:       valor ← atributos.get(i).getValues().get(j)
8:       se eNumero(valor) então
9:         numero ← numero + 1
10:      fim se
11:      se valoresUnicos.naoContem(valor) então
12:        valoresUnicos.adicionar(valor)
13:      fim se
14:    fim para
15:    se numero = atributos.get(i).getValues().size() então
16:      atributos.get(i).setType(numero)
17:    senão se valoresUnicos.size() <= maxDistinctValuesForNominal então
18:      atributos.get(i).setType(nominal)
19:    senão
20:      atributos.get(i).setType(string)
21:    fim se
22:  fim para
23: fim função

```

2.2. Interface Gráfica da Ferramenta

A interface gráfica foi construída utilizando-se a biblioteca gráfica Swing do Java. Na figura 3 pode-se visualizar a tela de uma consulta XQuery e o resultado da mesma. Na figura 4 visualiza-se o resultado do processo de automatização e as opções para exportação nos formatos ARFF e CSV.

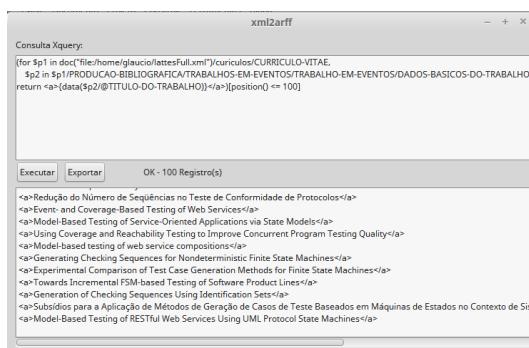


Figura 3. Tela de consulta

| Campo | Tipo | Valores |
|--------------------|---------|-----------------------|
| TITULO DO TRABALHO | string | pojo.Atribute@64377 |
| UF-NASCIMENTO | nominal | pojo.Atribute@345f... |

Seq. Valores

- 1 Steiner problem in graphs: Lagrange...
- 2 Lagrangian based heuristics for the...
- 3 Maximizing flow under special non...
- 4 A two-commodity flow approach to t...
- 5 Relax and Cut algorithm for the pr...
- 6 Optimal rectangular partitions
- 7 Complementary two-commodity flo...
- 8 A two-commodity flow approach for...
- 9 An optimization algorithm for mini...
- 10 Scheduling examinations to reduce ...
- 11 A cutting-planes approach to the St...
- 12 A cutting planes approach to the St...
- 13 A branch-and-cut algorithm for the...
- 14 A branch-and-cut algorithm for the...
- 15 Problema de Steiner em Grafos
- 16 Tight bounds for the Steiner proble...
- 17 Steiner Problem in Graphs: Lagrange...
- 18 A network flow based Lagrangean R...

Figura 4. Tela de automatização

3. Experimentos e Resultados

Realizou-se uma série de testes para mensurar o tempo médio de execução de reconhecimento de tipo de dados/frequência e tempo médio da exportação para o formato ARFF e CSV. Os testes foram realizados em um notebook com processador Intel core i5 de terceira geração com 4 núcleos de 2,9 Ghz, 8 GB de memória RAM e HD SSD de 240 GB. Na figura 5 pode-se visualizar os resultados obtidos para 100, 1000 e 10000 registros. Em cada etapa testou-se com 1, 5, 10, 15 e 20 atributos. Na figura 6 pode-se visualizar parte do arquivo ARFF gerado.

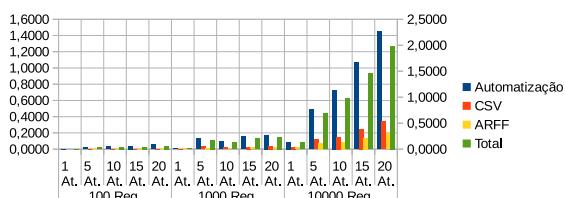


Figura 5. Tempo de cada etapa

```
%Generated with xq2arff - Xquery tool to generate
@RELATION producao
@attribute TITULO-DO-TRABALHO string
@attribute UF-NASCIMENTO {PI,PR}
@DATA
"Steiner problem in graphs: Lagrangean relaxation
"LAGrangian based heuristics for the linear order
```

Figura 6. Arquivo arff gerado

Observa-se que mesmo no pior caso(20 atributos e 10000 registros) o tempo total ficou próximo de 2 segundos(escala direita), portanto a ferramenta apresenta tempo de execução satisfatório quando comparado à extração/formatação manual. A validação do arquivo gerado foi realizada diretamente nos softwares Weka e R.

4. Considerações Finais e Trabalhos Futuros

A ferramenta xml2arff encontra-se em estágio preliminar. Mesmo assim ela já possibilita a redução do tempo de conversão de dados no formato XML para os formatos ARFF e CSV quando comparado com a formatação manual. Como trabalhos futuros se pretende expandir o número de formatos suportados(Matlab, Scilab e Sci2). Pretendemos tornar o aplicativo disponível como *software opensource* e com suporte a vários idiomas.

Referências

- Gentleman, R., Ihaka, R., Bates, D., et al. (2009). The r project for statistical computing.
URL: <http://www.r-project.org/254>.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009).
The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.
- Holmes, G., Donkin, A., and Witten, I. H. (1994). Weka: A machine learning workbench.
In *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on*, pages 357–361. IEEE.
- Hornik, K., Buchta, C., and Zeileis, A. (2009). Open-source machine learning: R meets weka. *Computational Statistics*, 24(2):225–232.
- Hornik, K., Zeileis, A., Hothorn, T., and Buchta, C. (2007). Rweka: an r interface to weka. *R package version 0.3-4.*, *URL* <http://CRAN.R-project.org/package=RWeka>.
- Jaffar, J., Michaylov, S., Stuckey, P. J., and Yap, R. H. (1992). The clp (r) language and system. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 14(3):339–395.

Palestras convidadas

as Convidadas

| | |
|--|---------------------|
| Uma Visão Sobre o Mundo das Operadoras de Telecomunicações | 164 |
| <i>Christian Schneider</i> | |
| Como uma Operadora de Telecomunicações Trata o Grande Volume de Dados? | 165 |
| <i>Roberto Yukio Nishimura</i> | |
| Data science | 166 |
| <i>Marcos André Gonçalves</i> | |
| A Evolução dos Negócios Digitais e a Necessidade de Novos Modelos Analíticos | 167 |
| <i>Emerson Cechin</i> | |
| Análise de Dados de Movimento: Você já Pensou que Está Sendo Monitorado . | 168 |
| <i>Vania Bogorny</i> | |

per:palestra1

Uma Visão Sobre o Mundo das Operadoras de Telecomunicações

Christian Schneider

Sercomtel S.A. - Telecomunicações <christian.schneider@>>

Sobre o autor: Christian Schneider é graduado em Direito pelo Centro Universitário de Brasília (Uniceub) e em Relações Internacionais pela Universidade de Brasília (UnB). Possui pós-graduação em Análise de Informações pela Escola de Inteligência da Agência Brasileira de Inteligência (Abin), em Política e Estratégia pela Escola Superior de Guerra (ESG/UnB), e em Gestão Econômica do Meio Ambiente pela UnB. Schneider é servidor de carreira da Abin desde 1996. Na sua carreira profissional já exerceu os cargos de Ministro de Estado Interino da Integração Nacional, secretário de Desenvolvimento do Centro-Oeste e diretor de Controle e vice-presidente do Banco de Brasília (BRB), de onde traz experiência em gestão de sociedade de economia mista.

per:palestra2

Como uma Operadora de Telecomunicações Trata o Grande Volume de Dados?

Roberto Yukio Nishimura

Sercomtel Participações S.A. <roberto.nishimura@sercomtel.com.br>

Sobre o autor: Roberto Yukio Nishimura possui graduação em Tecnologia em Processamento de Dados pelo Centro de Estudos Superiores de Londrina, pós-graduado em Administração de Engenharia de Software pela UNOPAR e em Geoprocessamento pela UFPR. Foi analista de suporte técnico e DBA por 13 anos, coordenador de suporte técnico por 4 anos e gestor de tecnologia da informação por 11 anos. Atualmente, é coordenador responsável pelo grupo de inovação da Sercomtel e gerente de desenvolvimento de negócios, uma área nova na Sercomtel para a busca de novos produtos e serviços que possam agregar valor aos produtos e serviços já ofertados pela empresa. Também é professor, tendo atuado em ensino de graduação e pós-graduação em várias instituições, incluindo UNIFIL, UNOPAR/KROTON, SENAI-SC, UTFPR-Medianeira, UTFPR-Pato Branco e UNIVEL-Cascavel.

per:palestra3

Data science

Marcos André Gonçalves

DCC/UFMG <mgoncalv@dcc.ufmg.br>

Sobre o autor: Marcos André Gonçalves possui graduação em Ciência da Computação pela Universidade Federal do Ceará (1995), mestrado em Ciência da Computação pela Universidade Estadual de Campinas (1997), doutorado em Computer Science pela Virginia Polytechnic Institute and State University (Virginia Tech) e pós-doutorado pela Universidade Federal de Minas Gerais (2006). Atualmente é professor Adjunto da Universidade Federal de Minas Gerais. Recebeu diversos prêmios e homenagens ao longo de sua carreira. Atua na área de Ciência da Computação com ênfase em Recuperação de Informação, Bibliotecas Digitais e Banco de Dados. É atualmente Membro Afiliado da Academia Brasileira de Ciências, Bolsista de Produtividade do CNPq (nível 1-D) e Bolsista do Programa Pesquisador Mineiro da Fapemig. (texto extraído do Currículo Lattes)

per:palestra4

A Evolução dos Negócios Digitais e a Necessidade de Novos Modelos Analíticos

Emerson Cechin

SEBRAE Paraná <ECechin@pr.sebrae.com.br>

Sobre o autor: Emerson Cechin é especialista em Gestão de Projetos e de Portfólio, Gestão de Negócios, Coordenação de Programas e Projetos, Análise de Investimentos e Gestão Financeira. Também atua como Professor Universitário e de Cursos Profissionalizantes. É atualmente Coordenador do Programa de TI Software e Gestor do Programa Empresas de Alto Potencial, ambos no SEBRAE-PR.

per:palestra5

Análise de Dados de Movimento: Você já Pensou que Está Sendo Monitorado?

Vania Bogorny

INE/UFSC <vania.bogorny@ufsc.br>

Resumo: Estamos vivendo a era do movimento. Grandes volumes de dados do nosso movimento diário estão sendo gerados, armazenados e analisados constantemente. Pelo Facebook, registramos os lugares aonde estamos, com quem nos comunicamos, o que gostamos e o que pensamos. Pelo Twitter registramos aonde estamos e o que pensamos. Ao acessar o Google, registramos aonde estamos e o que buscamos. Pelo GPS do nosso celular registramos detalhadamente o caminho por onde passamos. Com o nosso carro, equipado com GPS, registramos os lugares que visitamos e os caminhos que percorremos. São tantas as fontes de coleta de dados do nosso movimento que a ciência está desenvolvendo diversos métodos para o armazenamento e análise/mineração destes dados. Esta palestra tem como objetivo mostrar uma breve comparação dos diferentes tipos de dado de movimento e o que vem sendo feito na área de análise de dados gerados por dispositivos móveis, principalmente GPS, quais são as pesquisas atuais e as perspectivas para os próximos anos.

Sobre o autor: Vania Bogorny é professora do Departamento de Informática e Estatística da Universidade Federal de Santa Catarina desde Julho de 2009. Possui doutorado (2006) e mestrado(2001) em Ciência da Computação pela Universidade Federal do Rio Grande do Sul e graduação (1995) em Ciência da Computação pela Universidade de Passo Fundo, tendo recebido da Sociedade Brasileira de Computação o prêmio de melhor tese de doutorado (2007). Em 2014 realizou pós-doutorado no INRIA Sophia Antipolis, França; em 2008 realizou pós-doutorado no II/UFRGS e em 2007 realizou pós-doutorado na Universidade de Hasselt, Bélgica, no contexto do projeto europeu GeoPKDD, financiado pela União Européia. Em 2012 editou um livro sobre seu tema de pesquisa atual (Introdução a Trajetórias de Objetos Móveis). Em 2010 ministrou tutorial no tema de sua pesquisa no segundo maior congresso internacional na área de mineração de dados (IEEE ICDM). Desde 2009 atua em projetos de pesquisa internacionais como MODAP e SEEK, financiados pela União Européia (sendo coordenadora pela UFSC) e projeto de cooperação internacional Brasil/Itália, financiado pelo CNPQ. Nestes projetos estabeleceu parcerias de pesquisa com o CNR de Pisa/Itália, Universidade Ca'Foscari de Veneza/Itália e Universidade de Piraeus/Grécia. (texto extraído do Currículo Lattes)

Minicursos

Minicursos

| | |
|---|-----|
| Mineração de Opiniões | 170 |
| <i>Karin Becker</i> | |
| Machine Reading the Web – Além do Reconhecimento de Entidades Nomeadas e Extração de Relações | 171 |
| <i>Estevam R. Hruschka Jr.</i> | |
| Tecnologias para Gerenciamento de Dados na Era do Big Data | 173 |
| <i>Victor Teixeira de Almeida e Vitor A. Batista</i> | |

er:minicurso1

Mineração de Opiniões

Karin Becker

Resumo: Mídias sociais tornaram-se chave como forma de disseminação de ideias, opiniões, crenças, emoções e posicionamentos sobre os mais diversos assuntos. A opinião pública permite a organizações definir estratégias que melhorem seus produtos e serviços, ou aumentem o sucesso e visibilidade das marcas, entidades ou causas que representam. A análise de sentimentos tem por objetivo identificar automaticamente a partir de textos sentimentos tais como opiniões, emoções ou posicionamentos. Este minicurso tem como objetivo apresentar os principais conceitos e técnicas utilizadas para esta tarefa, e ilustrar sua aplicação prática usando um estudo de caso. O foco principal será na mineração de opiniões. Mais especificamente abordaremos: a) a motivação para a área, os diferentes tipos de sentimentos, e suas aplicações, b) a fundamentação teórica e os principais conceitos; c) as funções básicas de processamento de linguagem natural necessárias; d) o processo de mineração de opinião e as abordagens de classificação de polaridade; e) o uso de ferramentas para mineração de sentimento; e f) ilustração através de um estudo de caso.

Sobre a autora: Atua como professora adjunta no Instituto de Informática da UFRGS, onde é orientadora credenciada (mestrado e doutorado) no Programa de Pós-Graduação em Ciência da Computação, o qual foi avaliado pela CAPES com a nota 7. Possui ampla experiência de pesquisa tanto na academia quanto na indústria, sobretudo nas áreas de mineração de dados, inteligência de negócio, e computação baseada em serviços. Seus projetos de pesquisa atuais na área de mineração envolvem mineração de serviços web para apoio à evolução, bem como mineração de opinião para monitoração da evolução do sentimento. É também grande entusiasta de métodos ágeis, cujas técnicas, além de ensinar, adota em seus projetos. Possui quase uma centena de trabalhos publicados, entre artigos em periódicos e anais de conferências, capítulos de livros, e organização de obras. Já apresentou tutoriais, minicursos e palestras, sobretudo na área de mineração de dados, em eventos nacionais e latino americanos. Atuou como presidente do comitê de programa e do steering committee do SBBD e do ERBD, e participa como membro do comitê de programa de inúmeras conferências nacionais e internacionais de relevância. Possui graduação pela UFRGS (1984), mestrado em Ciências da Computação pela UFRGS (1989) e doutorado no Institut D'informatique – Facultés Universitaires Notre-Dame de la Paix – Bélgica (1993). (Texto retirado do Currículo Lattes)

er:minicurso2

Machine Reading the Web – Além do Reconhecimento de Entidades Nomeadas e Extração de Relações

Estevam R. Hruschka Jr.

Resumo: A web é inundada com informações em vários formatos diferentes, incluindo dados semi-estruturados e não estruturados. Machine Reading é uma área de pesquisa com o objetivo de construir sistemas que possam ler informações em linguagem natural, extrair conhecimento e armazená-lo em bases (estruturadas) de conhecimento. Neste minicurso será explorada a ideia de ler automaticamente o web usando técnicas de Machine Reading. Quatro das abordagens mais bem sucedidas nesta linha (DBpedia, Yago, OIE e NELL) serão apresentadas e discutidas, incluindo princípios, sutilezas e resultados atuais de cada abordagem. Recursos on-line de cada abordagem serão explorados, bem como as direções futuras indicadas por cada projeto. Apesar de se concentrar principalmente nos quatro projetos mencionados, algumas outras contribuições independentes em Machine Reading para Web serão mencionadas, bem como dois outros projetos industriais, nomeadamente Google Knowledge Vault e IBM Watson. O minicurso é destinado a preparar os participantes para iniciar novos trabalhos de investigação nesta área, bem como para conhecer o estado da arte, os principais desafios e alguns dos caminhos futuros mais promissores, bem como recursos disponíveis.

Sobre o autor: Estevam Rafael Hruschka Júnior possui graduação em Ciência da Computação pela Universidade Estadual de Londrina (1994), mestrado em Ciência da Computação pela Universidade de Brasília (1997) e doutorado em Sistemas Computacionais de Alto Desempenho pela Universidade Federal do Rio de Janeiro (PEC/COPPE/UFRJ) (2003). Foi jovem pesquisador FAPESP, é pesquisador CNPq (PQ-2) desde 2008, professor associado da Universidade Federal de São Carlos e professor adjunto na Carnegie Mellon University em Pittsburgh, EUA, onde lidera (em conjunto com os professores Tom Mitchell e William Cohen) o projeto Read The Web (<http://rtw.ml.cmu.edu>), no qual o primeiro sistema computacional de aprendizado de máquina sem fim foi proposto e implementado e continua sendo investigado e desenvolvido. Tem experiência na área de Ciência da Computação, com ênfase em Aprendizado de Máquina, Mineração de Dados e atuando principalmente nos seguintes temas: aprendizado de máquina, aprendizado sem fim, modelos gráficos probabilísticos, modelos Bayesianos, algoritmos evolutivos e teoria dos grafos. É membro do comitê editorial dos periódicos Intelligent Data Analysis – IOS Press e Advances in Distributed Computing and Artificial Intelligence Journal (ADCAIJ). Tem experiência no desenvolvimento de projetos de cooperação internacional e nacional com universidades como Carnegie Mellon University (EUA), Stanford University (EUA), University of Washington (EUA), Josef Stefan Institute (Slovenia), Oregon State university (EUA), Universidade do Porto (Portugal), Tsinghua University (China), University of Waterloo (CAN), University of Winnipeg (CAN), University of Manitoba (CAN), além de

parcerias e colaborações com empresas nacionais e multinacionais como Google Inc. (EUA), Yahoo! Inc. (EUA), Bloomberg (EUA), BBN Inc. (EUA), CYC Corp. (EUA), Nokia/Microsoft (Brasil e EUA), IBM Research (Brasil), E-BIZ Solutions (Brasil) e Siena Idea (Brasil).

er:minicurso3

Tecnologias para Gerenciamento de Dados na Era do Big Data

Victor Teixeira de Almeida e Vitor A. Batista

Este minicurso pretende explorar e diferenciar de forma introdutória diversas tecnologias recentes para gerenciamento de dados na era do big data. Será utilizado como pano de fundo para os exemplos um problema clássico da comunidade de bancos de dados: a contagem de triângulos. Esta problema é perfeito para explicar as diferenças entre as tecnologias em termos de representatividade e desempenho, pois pode ser representado por uma simples consulta SQL com duas junções, contudo extremamente complexa de ser executada eficientemente. A partir deste problema, é possível identificar como cada tecnologia se comporta em representar sua solução, bem como seu desempenho. Demonstrações de algumas das principais tecnologias gratuitas serão feitas durante o minicurso.

Sobre o autor: Victor Teixeira de Almeida (ministrante) é analista de sistemas na Petrobras, atua na Arquitetura Tecnológica de TIC; e também professor adjunto 20h na Universidade Federal Fluminense (UFF). Mestre em bancos de dados pela COPPE/UFRJ; doutor em bancos de dados pela Universidade de Hagen, Alemanha; passou um ano (2013) como pós-doutor na Universidade de Washington, Seattle, EUA, no Projeto Myria, uma plataforma de big data como serviço na nuvem. Especialista no gerenciamento de grandes volumes de dados e tecnologias de big data.

Sobre o autor: Vitor A. Batista possui Mestrado e Graduação em Ciência da Computação pela Universidade Federal de Minas Gerais. Possui também duas pós-graduações Lato-sensu em Gestão de Negócios e Finanças pela Fundação Dom Cabral. Tem trabalhado com desenvolvimento de software desde 1997, utilizando diversas tecnologias e frameworks. Atuou como analista de negócios e requisitos, implementador e gerenciou projetos de sistemas de médio e grande porte. No passado foi gerente de processos do Synergia Engenharia de Software e Sistemas (UFMG). Atualmente trabalha como Engenheiro de Software na Petrobras, avaliando novas tecnologias que possam contribuir com a inovação de processos na Companhia. Lecionou diversos cursos de graduação e pós-graduação e possui diversos artigos publicados em conferências e periódicos de Engenharia de Software. Possui certificações técnicas do IEEE (Certified Software Development Professional), da Microsoft (MCSD, MCSE) e da OMG (UML Certified Professional).

Sumário (Oficinas)

Oficinas

| | |
|--|-----|
| Oficina Para Meninas | 175 |
| <i>Nádia Puchalski Kozievitch e Silvia Amélia Bim</i> | |
| Redes Complexas e Processamento de Grafos na era do Big Data | 177 |
| <i>Luiz Celso Gomes Júnior</i> | |
| Usando dados de mídia social para o entendimento de sociedades urbanas | 178 |
| <i>Thiago Henrique Silva</i> | |

Oficina para Meninas

Nádia Puchalski Kozievitch e Silvia Amélia Bim

Resumo: Esta oficina tem como objetivo investigar a curiosidade sobre o tema Banco de Dados em alunas do Ensino Médio. A intenção é divulgar a área de Computação para despertar o interesse de estudantes do Ensino Médio/Tecnológico ou dos anos finais do ensino fundamental, para que conheçam melhor a área e, desta forma, motivá-las a seguir carreira em Computação, que históricamente tem sido predominantemente escolhida pelo público masculino. Nesta primeira abordagem não é adotada nenhuma aplicação em específico, o único pré-requisito necessário para as participantes é possuir conhecimento básico de computadores e internet. A metodologia usada na oficina é composta pelas seguintes etapas: Um questionário inicial, a apresentação da oficina, e um questionário final. Serão abordados os seguintes tópicos: (i) Conceitos Básicos de Banco de Dados: BD, SGBD, Modelo Relacional, etc.; (ii) Motivação: ilustração de diferentes aplicações dentro da área de Banco de Dados, como as Bibliotecas Digitais: Biblioteca Digital da Universidade de Kabul, Biblioteca Digital sobre Chopin, entre outros; (iii) Tipos de Bancos de Dados; (iv) Ideia Básica de Otimização; e (v) Exemplo simples de uso de Bancos de Dados (como o Mapeamento de Rios, represas, áreas indígenas e nascentes da COPEL). Em paralelo, alguns exercícios serão propostos. Dentre o material utilizado, buscar-se-á focar em um impacto visual, em uma integração com temas atuais (como redes sociais, YouTube, etc.), em aplicações atuais, e em possibilidades de continuar o aprendizado (em fontes externas, como banco de dados e aplicações para crianças, tutoriais de SQL, entre outros). Dentre as dificuldades e desafios que esperamos enfrentar, podemos citar: (i) o tratamento de temas teóricos de Banco de Dados para instigar alunas do ensino médio; (ii) a integração de equipes diferenciadas na problemática; (iii) a integração de conteúdos dinâmicos da Web (Sites, Redes Sociais, etc.) para atrair a atenção das alunas; e (iv) a ilustração de como aplicações atuais (Facebook, YouTube, etc.) se baseiam em banco de dados e computação.

Sobre a autora: Nádia Puchalski Kozievitch possui graduação em Ciências da Computação pela Universidade Federal do Paraná (2001), mestrado em Informática pela Universidade Federal do Paraná (2005) e doutorado em Ciências da Computação pela Universidade Estadual de Campinas (2011). No período de fevereiro/2010 a setembro/2010 fez doutorando sanduíche, no Digital Library Research Laboratory (DLIB), na Virginia Polytechnic Institute and State University (EUA). Trabalhou em projetos de P&D na área de telefonia na IBM (2006-2012); e na Companhia Paranaense de Energia (Copel/Simepar), na área de meteorologia (1999 -2004). Atualmente é professora efetiva da Universidade Tecnológica Federal do Paraná (UTFPR), câmpus Curitiba. Atua como professor permanente no Programa de Pós-Graduação em Computação Aplicada (PPGCA, UTFPR). Tem experiência na área de Ciência da

Computação, com ênfase em Banco de Dados. Seus interesses englobam bibliotecas digitais, GIS e recuperação de informação baseada em conteúdo.

Sobre a autora: Silvia Amélia Bim é bacharel em Ciência da Computação pela Universidade Estadual de Maringá (1998), mestre em Ciência da Computação pela Universidade Estadual de Campinas (2001) e doutora em Ciências – Informática pela Pontifícia Universidade Católica do Rio de Janeiro (2009). Atualmente é professora adjunta da Universidade Tecnológica Federal do Paraná (UTFPR), no campus de Curitiba. É secretária adjunta da Regional Paraná da Sociedade Brasileira de Computação (SBC) e coordenadora do Programa Meninas Digitais (SBC). Também coordena o projeto de extensão Emíli@s – Armação em Bits na UTFPR-CT. Suas áreas de interesse são: Interação Humano-Computador (IHC), Engenharia Semiótica, Avaliação de Interfaces, Método de Inspeção Semiótica (MIS), Método de Avaliação de Comunicabilidade (MAC), Ensino de IHC e Mulheres na Computação.

aper:oficina2

Redes Complexas e Processamento de Grafos na era do Big Data

Luiz Celso Gomes Júnior

A oficina abordará a análise e processamento de Grafos e Redes Complexas, sob o ponto de vista teórico e prático. Discutiremos as definições, aplicações e principais algoritmos das áreas. Desenvolveremos também exercícios didáticos, cobrindo desde a obtenção e manipulação dos dados, passando pelo armazenamento e consultas em Bancos de Dados de Grafos, até a análise e visualização das redes. Abordaremos também aspectos de desempenho, analisando as tecnologias que permitem o processamento distribuído de volumes massivos de dados interconectados.

Sobre o autor: Professor na UTFPR, doutor em Ciência da Computação pela UNICAMP. Pesquisador em Bancos de Dados há 10 anos, atuando também nas universidades de Waterloo (Canadá) e UPMC (França). Nos últimos anos tem focado em tópicos relacionados à análise, armazenamento e gerenciamento de Redes Complexas.

aper:oficina3

Usando dados de mídia social para o entendimento de sociedades urbanas

Thiago Henrique Silva

A popularização de dispositivos portáteis, como smartphones, assim como a adoção mundial de sites de mídia social permitem cada vez mais a um usuário estar conectado e compartilhar dados de qualquer lugar, a qualquer momento. Nesse cenário, as pessoas participam como sensores sociais, fornecendo dados voluntariamente que capturam as suas experiências de vida diária. Um dos principais objetivos desta oficina é mostrar que dados de mídias sociais (e.g., Instagram e Foursquare) podem atuar como valiosas fontes de sensoriamento em larga escala, proporcionando acesso a características importantes da dinâmica de cidades e do comportamento social urbano, de forma rápida e abrangente. Esta oficina discutirá como trabalhar com dados de mídia social, analisando as suas propriedades e a sua utilidade no desenvolvimento de aplicações mais sofisticadas em diversas áreas. Além disso, alguns dos principais desafios e oportunidades de pesquisa relacionados serão discutidos.

Sobre o autor: Thiago H Silva é professor do Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná em Curitiba. Em 2014, recebeu o título de doutor em Ciência da Computação pela Universidade Federal de Minas Gerais, com doutorado sanduíche na University of Birmingham, Reino Unido (2012), e no INRIA-Paris, França (2013). Em 2011, foi um pesquisador visitante por seis meses na Telecom Italia, Itália. Ele recebeu, em 2009, o título de mestre em Ciência da Computação pela Universidade Federal de Minas Gerais. Foi agraciado com os prêmios de melhor trabalho/menção honrosa nos seguintes eventos: Concurso de Teses e Dissertações da SBC (2015), Semana de Seminários de Teses do DCC-UFMG (2013), IEEECPSCOM (2012), SBRC (2009, 2010 e 2013) e SBCUP (2012).

ERBD

XII ESCOLA REGIONAL DE BANCO DE DADOS

LONDRINA 2016

TEMA DATA SCIENCE

[HTTP://CROSS.DC.UEL.BR/ERBD2016](http://cross.dc.uel.br/ERBD2016)



PROMOÇÃO



REALIZAÇÃO



PATROCÍNIO

OURO



PRATA



Apoio ao Desenvolvimento Científico
e Tecnológico do Paraná

BRONZE



APOIO

