ERBD PASSO FUNDO 2017

XIII ESCOLA REGIONAL DE BANCO DE DADOS

TEMA: SISTEMAS DE RECOMENDAÇÃO

ANAIS

XIII ESCOLA REGIONAL DE BANCO DE DADOS
http://sbc.org.br/erbd2017/

03 a 05 de abril de 2017
Passo Fundo – RS – Brasil

ANAIS

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

Organização
Universidade de Passo Fundo – UPF

Comitê Diretivo da ERBD
Daniel Luís Notari – UCS (Presidente)
Carmem Satie Hara – UFPR
Daniel dos Santos Kaster – UEL

Chair Local
Cristiano Roberto Cervi – UPF (Coordenador Geral)

Comitê de Programa
Daniel Luís Notari – UCS

ISSN: 2177-4226

Dados Internacionais de Catalogação na Publicação (CIP)

E74 Escola Regional de Banco de Dados (13. : 2017 : Passo Fundo, RS)
Sistemas de recomendação [recurso eletrônico] / [coordenação geral] Cristiano Roberto Cervi. – Passo Fundo : Ed. Universidade de Passo Fundo, 2017.

Modo de acesso: <http://www.sbc.org.br/erbd2017>
ISSN: 2177-4226

1. Banco de dados – Congressos e convenções. 2. Banco de dados. I. Cervi, Cristiano Roberto, coord. II. Anais da XIII Escola Regional de Banco de Dados .

CDU: 519.68.023

Bibliotecária responsável Marciéli de Oliveira - CRB 10/2113

Editorial

É com grande satisfação que apresentamos os artigos aceitos para a décima terceira edição da Escola Regional de Banco de Dados (ERBD) e que compõem os anais do evento. Em 2017, a ERBD ocorreu de 03 a 05 de abril, na cidade de Passo Fundo-RS, sob a organização da Universidade de Passo Fundo (UPF).

A ERBD é um evento anual promovido pela Sociedade Brasileira de Computação (SBC),por intermédio da sua Comissão Especial de Banco de Dados (CEBD). O evento tem por 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. O tema da ERBD 2017 foi Sistemas de Recomendação devido a sua multidisciplinaridade e relevância atual. Estes têm por objetivo sugerir/recomendar itens de um determinado domínio que melhor se relacionam às preferências ou interesses do usuário.

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. Além das sessões técnicas, a programação do evento contou com oficinas, minicursos, palestras e painéis proferidas por pesquisadores de renome da comunidade brasileira.

A categoria de Pesquisa recebeu 13 submissões, das quais 7 foram aceitas, o que representa 54% 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 13 foram aceitas, o que representa 86% de taxa de aceitação. Artigos desta categoria foram apresentados em 10 minutos nas sessões técnicas, bem como na forma de pôster.

Os Anais da XIII ERBD representam o resultado do esforço coletivo de um grande número de pessoas. Agradecemos ao Comitê de Organização Local da ERBD, coordenados pelos Prof. Juliano Tonezer da Silva (categoria Pesquisa) e Prof. Victor Billy da Silva (categoria Aplicações/Experiências), que trabalharam arduamente para garantir o bom andamento do evento. Gostaríamos de agradecer também aos membros do Comitê de Programa que fizeram revisões de excelente qualidade.

Por fim, agradecemos aos autores que submeteram seus trabalhos para a XIII ERBD.

Dr. Daniel Luis Notari, UCS
Coordenador do Comitê de Programa da Categoria Pesquisa

Dra. Helena Graziottin Ribeiro, UCS
Coordenadora do Comitê de Programa da Categoria Aplicações/Experiências

CARTA DO COORDENADOR GERAL

Movidos por um grande desafio, realizamos mais uma edição da Escola Regional de Banco de Dados na Universidade de Passo Fundo, repetindo a dose de 2005 (na época, a II edição da escola). Nesta edição de 2017, a XIII ERBD contou com aproximadamente 300 participantes, incluindo alunos de graduação e pós-graduação, bem como diversos profissionais da indústria da computação de Passo Fundo e região sul do país. O tema Sistemas de Recomendação foi o escolhido para articulação das palestras, minicursos, oficinais, painéis e apresentação de artigos de pesquisa e de aplicações/ferramentas.

Ao todo, 25 palestrantes colaboraram com suas experiências acadêmicas e profissionais, proporcionando um evento de altíssimo nível e contribuindo para a atualização e qualificação do público que prestigiou o evento.

O grupo de professores e alunos envolvidos na organização da Escola foi fundamental para que a ERBD 2017 fosse executada com êxito. A todos, nosso muito obrigado pela disponibilidade, responsabilidade e entusiasmo para ajudar, de forma voluntária e colaborativa, para que nossos objetivos fossem alcançados.

Nosso agradecimento ao Comitê Diretivo da ERBD, à Comissão Especial de Banco de Dados (CEBD) e à direção e colaboradores da Sociedade Brasileira de Computação por todo o apoio prestado para que conseguíssemos realizar mais uma edição da escola regional.

Por fim, nosso agradecimento especial à Universidade de Passo Fundo e ao Instituto de Ciências Exatas e Geociências, por terem cedido suas estruturas e pessoal técnico-administrativo para todo o suporte necessário à realização do evento.

Dr. Cristiano Roberto Cervi
Coordenador Geral da ERBD 2017

XIII ESCOLA REGIONAL DE BANCO DE DADOS

http://sbc.org.br/erbd2017/

03 a 05 de abril de 2017
Passo Fundo – RS – Brasil

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

Organização
Universidade de Passo Fundo – UPF

Comitê Diretivo da ERBD
Daniel Luís Notari – UCS (Presidente)
Carmem Satie Hara – UFPR
Daniel dos Santos Kaster – UEL

Comissão Organizadora
Coordenação Geral – Cristiano Roberto Cervi (UPF)
Comitê de Programa – Daniel Luis Notari e Juliano Tonezer da Silva (UPF)
Palestras – José Maurício Carré Maciel (UPF) e Karin Becker (UFRGS)
Minicursos – Eder Pazinatto (UPF)e Ronaldo dos Santos Mello (UFSC)
Oficinas – Jaqson Dalbosco (UPF) e Daniel dos Santos Kaster (UEL)
Aplicações/Experiências – Victor B. da Silva (UPF) e Helena Graziottin Ribeiro (UCS)
Sessões Técnicas – Juliano Tonezer da Silva (UPF) e Daniel Luis Notari (UCS)
Painéis – Rafael Rieder (UPF) e Carmem Satie Hara (UFPR)
Infraestrutura e Logística – Marcos José Brusso (UPF) e Gilberto Gampert (UPF)
Articulação com Empresas – Alexandre Lazaretti Zanatta (UPF); Luiz F. Amaral e Silva (PoloSul.org) e Ary Cover (APLTEC)

Comitê de Programa
Alcides Calsavara (PUCPR)
André Schwerz (UTFPR)
Angelo Frozza (IFC)
Carina F. Dorneles (UFSC)
Carmem S. Hara
Cristiano R. Cervi (UPF)
Daniel Kaster (UEL)
Daniel Notari (UCS) – Coordenador
Deborah Carvalho (PUCPR)
Deise Saccol (UFSM)
Denio Duarte (UFFS)
Eder Pazinatto (UPF)
Edimar Manica (IFRS)
Eduardo Borges (FURG)

Eduardo Cunha de Almeida (UFPR)
Fernando José Braz (IFC)
Flavio Uber (UEM-UFPR)
Guilherme Dal Bianco (UFFS)
Guillermo Nudelman Hess (FEEVALE)
Gustavo Zanini Kantorski (UFSM)
Helena Ribeiro (UCS)
Joao Marynovski (PUCPR)
José Maurício Carré Maciel (UPF)
Karin Becker (UFRGS)
Luiz Celso Gomes Jr (UTFPR)
Marcos Aurélio Carrero (UFPR)
Raqueline Penteado (UEM-UFPR)
Rebeca Schroeder (UDESC)
Regis Schuch (UFSM)
Renata Galante (UFRGS)
Renato Fileto (UFSC)
Ronaldo Mello (UFSC)
Sandro Camargo (UNIPAMPA)
Scheila de Ávila e Silva (UCS)
Sergio L. S. Mergen (UFSM)
Solange de Lurdes Pertile (UFSM)
Vania Bogorny (UFSC)

Sumário

Artigos Completos de Pesquisa

Uma Proposta de Abordagem de Recomendação para Carreira de Pesquisadores Baseada em Personalização, Similaridade de Perfil e Reputação Acadêmica

Gláucio R. Vivian[1], **Cristiano R. Cervi**[1]

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

{149293,cervi}@upf.br

Abstract. *The Recommendation Systems seek to suggest relevant information to users. In the context of researchers there are numerous approaches proposed to recommend articles and citations. The objective of this work is to present a strategy of recommendation approach focused on the career projection of researchers. As a result, we obtained the best combination for subarea profile similarity with Log-likelihood and Apache Mahout's ClassicAnalyzer class. Regarding the recommendations, two different types of recommendation were generated for several test groups. The results were satisfactory and show that the proposed approach has good coverage in the generation of recommendations.*

Resumo. *Os Sistemas de Recomendação procuram sugerir informações relevantes aos usuários. No contexto dos pesquisadores existem inúmeras abordagens propostas para recomendar artigos e citações. O objetivo deste trabalho é apresentar uma estratégia de abordagem para recomendação com foco na projeção da carreira de pesquisadores. Como resultados, obtivemos a melhor combinação para similaridade de perfil de subárea com Log-likelihood e a classe ClassicAnalyzer do Apache Mahout. Com relação as recomendações, foram geradas dois tipos diferentes de recomendação para diversos grupos de testes. Os resultados foram satisfatórios e demostram que a abordagem proposta tem boa cobertura na geração de recomendações.*

1. Introdução

Os Sistemas de Recomendação (SR) tradicionais buscam auxiliar os usuários na seleção de conteúdos. No campo da pesquisa científica, a realidade dos pesquisadores está convergindo para um aumento significativo na quantidade e diversidade de produção. Além das tradicionais publicações no formato de artigos científicos, existem inúmeras outras formas de produção que aos poucos estão sendo estimuladas. Dentre muitas, podem ser citadas: patentes, *softwares*, orientações, revisões, editoração, livros, projetos de pesquisa e rede de colaboração. Este novo paradigma imposto aos pesquisadores, torna mais complexa e árdua a tarefa de traçar planos estratégicos para projeção da carreira do pesquisador. Neste contexto, os Sistemas de Recomendação podem interagir com os pesquisadores, buscando orientá-los com estratégias de recomendações no planejamento da sua carreira. Em outras palavras, um Sistema de Recomendações pode sugerir ao pesquisador o que, como e quando realizar determinada produção. Como resultado, tem-se a possibilidade de estar realizando a atividade mais adequada e na ordem cronológica mais apropriada.

O objetivo deste trabalho é apresentar uma abordagem de recomendação baseada na personalização dos dados de pesquisadores, usando a similaridade de perfil e reputação acadêmica como premissa de recomendação. A abordagem proposta visa contribuir para o planejamento da carreira do pesquisador, bem como ser um apoio a grupos de pesquisa, programas de pós-graduação e instituições, para que acompanhem a evolução da vida científica de um pesquisador. A proposta vem ao encontro com a necessidade de otimizar recursos humanos e financeiros. Além disso, possibilita um incremento no desenvolvimento científico e tecnológico por meio do aumento da produtividade de forma qualificada. Desse modo, garante-se que o planejamento esteja alinhado com a realidade atual de alta oferta de informações, que demanda precisão e agilidade para identificar as tendências do cenário onde o pesquisador está inserido.

Este artigo está organizado da seguinte forma: Na seção 2 são analisados alguns trabalhos correlatos. Na seção 3 é exposta a abordagem proposta. Na seção 4 é apresentada a metodologia. Na seção 5 são apresentados os experimentos e resultados encontrados. Finalmente, na seção 6, são apresentadas as conclusões e trabalhos futuros.

2. Trabalhos Correlatos

Esta seção apresenta trabalhos correlatos existentes no contexto de Sistemas de Recomendação para pesquisadores.

O artigo de [Middleton et al. 2004] introduziu a modelagem de perfil para recomendação de artigos científicos com o uso de ontologias. A representação dos artigos foi realizada utilizando-se vetores de termos, computados com a técnica *Term Frequency* (TF) e divididos pelo total de termos. Os experimentos demostraram que a abordagem proposta supera os sistemas apresentados na literatura.

No trabalho de [Ekstrand et al. 2010] foram explorados diversos métodos (177 algoritmos em 5 famílias) para recomendação de artigos científicos baseada em filtragem colaborativa e em conteúdo. O perfil do usuário foi construído com as próprias citações da Web. As medidas de influência foram realizadas com os algoritmos HITS[Kleinberg 1999] e *PageRank*[Page et al. 1999]. Inicialmente, realizou-se testes *offline*, posteriormente se conduziu uma avaliação *online* com pesquisadores. Ao final, demostrou-se que os usuários preferem as recomendações com filtragem colaborativa.

No trabalho de [Zhang e Li 2010], foi proposta a recomendação de artigos com o modelo de perfil baseado em árvores para ultrapassar os inconvenientes do espaço vetorial. A abordagem proposta cria o perfil do pesquisador com base nos trabalhos visualizados. A correlação entre os perfis é computada utilizando a técnica *Edit Distance* adaptada para árvores. Um modelo de ativação disperso é construído para localizar perfis com interesses semelhantes. Para avaliar foi utilizada a métrica *Normalized Discounted Cumulative Gain* com um subconjunto com 60 mil exemplares da *National Science and Technology Library*. Foram realizadas avaliações de 5 até 30 recomendações. Ao final, a melhor opção foi a recomendação de 10 artigos.

A proposta de [Huang et al. 2012] consiste em definir citações usando palavras explícitas no texto. Posteriormente é proposto um modelo baseado em um dicionário que contem a probabilidade de translação de uma dada referência em uma palavra ou frase para todos os termos da linguagem descritiva. Em seguida é computada a probabilidade

de uma dada referência em questão usando as probabilidades da translação. Finalmente as referências passam por um *ranking* e recomenda-se as 20 primeiras. Ao final dos experimentos, os autores afirmam que a proposta ultrapassa o estado atual da arte.

No trabalho de [Beel et al. 2013] foi feita uma revisão sistemática de 80 abordagens existentes para recomendar artigos científicos. Constatou-se que existem mais de 170 artigos publicados. Após a análise, foi constatado que 21% não foram avaliados. Entre os avaliados, cerca de 19% não foram avaliados em relação ao *baseline*. Com relação ao tipo de avaliação, somente 5 trabalhos (7%) foram avaliados de forma *online*. A maioria dos experimentos avaliativos (cerca de 69%) foi realizada de forma *offline*. As fontes para avaliações foram obtidas do CiteSeer (29%), ACM (10%), e CiteULike (10%). Ao final foi concluído que não é possível identificar qual abordagem é mais promissora, pois não existe um consenso de qual trabalho representa o estado da arte.

Nos trabalhos de [Sugiyama e Kan 2013, Sugiyama e Kan 2015] buscou-se construir um sistema de recomendações por meio do potencial de citação de artigos. Foi construído um perfil vetorial com base nos artigos publicados na DBLP e na ACM *Digital Library*. Utilizou-se a correlação de Pearson entre o perfil e vetor para recomendar os artigos com maior similaridade para os usuários alvo. Após diversos experimentos com o objetivo de ajustar a acurácia em 10% com relação ao *baseline*, os autores afirmam que a abordagem proposta é eficiente em caracterizar artigos para recomendação.

Como se observou nesta seção, existem diversos trabalhos com o objetivo de auxiliar os pesquisadores com recomendações de artigos, referências e citações. Contudo não localizamos estudos com a finalidade de recomendações para o planejamento de carreira de pesquisadores. A abordagem proposta busca preencher esta lacuna identificada.

3. Abordagem Proposta

Nesta seção definimos uma abordagem com o intuito de gerar recomendações para o planejamento de carreira de pesquisadores. A abordagem proposta para gerar as recomendações deve responder aos seguintes questionamentos: **i) O que Fazer?** Recomendar o que os pesquisadores com maior reputação da mesma subárea (consonância com o que produz) realizaram. Em outras palavras, essa abordagem sugere que se siga os passos de outros pesquisadores com mais prestígio na mesma subárea de atuação. **ii) Como Fazer?** Descrição de como realizar a atividade recomendada apresentando opções para o pesquisador. **iii) Quando Fazer?** Fazer por primeiro o que tiver maior impacto na reputação do pesquisador para que ele possa evoluir na carreira. A abordagem proposta deve apresentar ao pesquisador os itens recomendados em ordem decrescente de relevância para a reputação do mesmo.

A abordagem proposta vai ser divida em duas partes: **i) Recomendações Não Personalizadas:** são as que não possuem informações específicas para o usuário, elas se caracterizam por apenas considerarem o elemento do Rep-Index e sua importância para o aumento da reputação. **ii) Recomendações Personalizadas:** são específicas para cada usuário, a similaridade de perfil e reputação dos demais pesquisadores são utilizados para gerar a recomendação.

A etapa inicial consiste na adaptação do modelo do perfil do pesquisador. Optou-se por realizar esta tarefa no Rep-Model proposto por [Cervi et al. 2013b,

Cervi et al. 2013a]. Ele é um conjunto de elementos que representam o comportamento acadêmico e científico dos pesquisadores. O Rep-Model também é utilizado no Rep-Index, trata-se de um índice para classificar pesquisadores com outros critérios além de artigos e citações. Na proposta está incluído grau de instrução, bancas, orientações, comitês e produção. O grande diferencial de outras métricas é a média ponderada, escopo abrangente e adaptabilidade. As alterações propostas incluem novos elementos no Rep-Model com o intuito de utilizá-lo para esta finalidade. Na Tabela 1 pode-se visualizar a proposta de adição de elementos ao Rep-Model.

Tabela 1. Elementos adicionados ao Rep-Model.

Rep-Model Original			**Adições ao Rep-Model**	
Port.	**Ing.**	**Elemento**	**Port.**	**Elemento**
NM	NM	Nome	CP	Cultivar Protegida
INST	INST	Instituição	CR	Cultivar Registrada
GI	ED	Grau de Instrução	DI	Desenho Industrial
OP	PA	Orientação de Pós-doutorado	MARC	Marca
OD	PTA	Orientação de Doutorado	PAT	Patente
OM	MDA	Orientação de Mestrado	TCI	Topografia Circuíto Integrado
PBM	PEBPT	Participação em Banca de Mestrado	PRODTEC	Produto Tecnológico
PBD	PEBMD	Participação em Banca de Doutorado	PROCTEC	Processo ou Técnicas
MCEP	EBM	Membro de Corpo Editorial de Periódico	TT	Trabalho Técnico
RP	RJ	Revisão de Periódico	PREM	Prêmios
CCC	CCC	Coordenação de Comitê de Conferência		
MCC	CCM	Membro de Comitê de Conferência		
AP	ASJ	Artigo em Periódico	**Adições Textuais ao Rep-Model**	
LIV	BP	Livro	TPB	Títulos Produção Bibliográfica
CLIV	BCP	Capítulo de Livro	TPT	Títulos Produção Técnica
TCC	CWPCP	Trabalho Completo em Conferência	RC	Resumo do Currículo
HI	HI	H-Index	AA	Áreas Atuação
RC	NC	Rede de Coautoria	TO	Títulos de Orientações
PP	RP	Projeto de Pesquisa	TB	Títulos de Bancas
SOFT	SOFT	Software	TOA	Títulos Orientações Andamento

A adição de novos elementos quantitativos ao Rep-Model tem a finalidade de contemplar a diversidade de produção anteriormente mencionada. Quanto aos elementos textuais, os mesmos são utilizados para construir um perfil de subárea de atuação e são importantes para localizar as afinidades entre os pesquisadores. Optou-se por utilizar a(s) subárea(s) de atuação da plataforma Lattes devido ao fato das mesmas serem mais específicas e representarem a(s) sua(s) área(s) de atuação.

O próximo passo da abordagem é a definição da utilização dos elementos definidos para o modelo de perfil (Rep-Model adaptado). Os elementos do tipo inteiro do Rep-Model são utilizados para computar o Rep-Index. Os resultados de cada usuário são armazenados em uma posição do vetor $Rep - \vec{I}ndex$. Os elementos textuais, exceto NN e INST, são submetidos a uma etapa mais complexa que os quantitativos. Nesta fase é inferido um perfil com base nas informações textuais. Para isso, optou-se por empregar técnicas de recomendação baseada em conteúdo. Na Figura 1 pode-se visualizar as etapas do processo de *Text mining*.

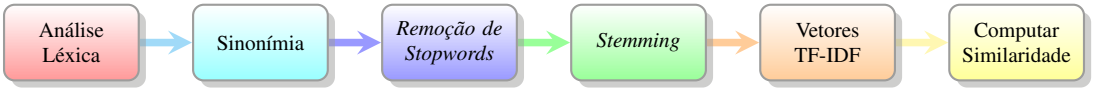

Figura 1. Etapas do processo de *Text Mining*.

A primeira etapa é a análise léxica que tem por objetivo separar as informações em palavras. A segunda etapa trata-se da sinonímia, a mesma busca por meio de um dicionário de sinônimos aproximar textos semanticamente semelhantes. Em seguida ocorre a remoção de *stopwords*, ela busca a eliminação de listas de classes de palavras sem

relevância ou que podem gerar falsas similaridades. Posteriormente, utiliza-se a técnica denominada de *stemming*, a mesma procura reduzir as palavras ao seu radical por meio da supressão de sufixos. Finalmente, aplica-se a técnica de vetorização denominada TF-IDF (*Term Frequency-Inverse Document Frequency*). A mesma simplifica o processamento das informações textuais por meio da representação em vetores esparsos com a frequência de ocorrência e relevância dos seus termos.

O resultado da técnica de vetorização é aplicado às funções de correlação de Pearson, Spearman e Kendall Tau; similaridade Fuzzy; distância Euclidiana, Canberra, Tanimoto, Log-likelihood, Manhattan, Minkowski, Chebyshev, Coseno e EarthMovers. Cada uma dessas funções terá como resultado final uma matriz triangular $M_{sim_{(U_{m,n})}}$ onde são armazenadas as similaridades entre os perfis dos pesquisadores. No caso das distâncias os valores foram todos convertidos em similaridades (normalizados) por meio da Equação: $s = \frac{1}{1+d}$. Onde: d representa a distância e s a similaridade obtida no intervalo de valores entre 0 e 1, inclusive.

As **recomendações personalizadas** sobre "o que fazer" para um usuário U são realizadas pela análise do vetor $Rep - \vec{I}ndex_{U(i)}$, onde são localizados os pesquisadores que possuem maior reputação que U. A matriz $M_{sim_{(U_{m,n})}}$ também é utilizada para localizar os perfis mais semelhantes com relação a subárea de atuação. A partir do conjunto de n possíveis recomendações ao usuário U, deve-se computar o quanto cada uma incrementa na sua reputação.

As **recomendações não personalizadas** são geradas a partir da simulação do aumento do Rep-Index do pesquisador em questão. O aumento da reputação (denotado por Δ) para um pesquisador pode ser calculado pela diferença entre a nova reputação e a sua atual. A nova reputação deve ser computada considerando o incremento hipotético de uma unidade no elemento desejado. O cálculo do aumento da reputação pode ser realizado pela equação: $\Delta_{(R)} = Rep - Index\ Novo_{(R)} - Rep - Index\ Atual_{(R)}$. Esta equação é funcional para a maioria das situações, contudo não considera a situação do valor máximo (teto) do elemento em questão. Além deste fato, existe a necessidade de computar todos os elementos do Rep-Index para obter a reputação nova e atual. Pode-se simplificar a mesma e corrigir a situação acima mencionada. A Equação 1 apresenta uma proposta melhorada para o cálculo em questão.

$$\Delta_{(R)} = \begin{cases} 0, & \text{se } inc \geq max_{(R_i)} \\ \frac{inc * w_{(R_i)}}{max_{(i)}}, & \text{senão} \end{cases} \quad (1)$$

Onde, $\Delta_{(R)}$ é o aumento na reputação do pesquisador R, inc representa o incremento desejado para o elemento, i indica o elemento do Rep-Index em questão para o pesquisador R, e $max_{(i)}$ é o valor máximo do elemento i no grupo formado pelos pesquisadores do CNPq para a área em questão. Observa-se que o aumento da reputação é diretamente proporcional ao peso do elemento e inversamente proporcional ao valor máximo do elemento para o grupo dos pesquisadores do CNPq da área.

4. Metodologia

Para realização dos experimentos, foram utilizados dados de pesquisadores da área da Ciência da Computação. Os dados foram coletados da plataforma Lattes[1] e Google Scholar[2] entre novembro de 2016 e janeiro de 2017. No trabalho de [Vivian e Cervi 2016a] pode-se encontrar detalhadamente os passos para recuperação das informações e criação do XML *Dataset* utilizado para realizar os experimentos. As consultas dos dados foram realizadas com auxílio da linguagem XQuery por meio do *software* Basex[3]. O intercâmbio de informações entre formatos foi realizado utilizando-se o *software* Xml2Arff[4] proposto por [Vivian e Cervi 2016b].

A avaliação da abordagem proposta foi realizada com os seguintes experimentos: **i)** Encontrar o melhor conjunto de sinonímia, *Stopwords* e *Stemming*. **ii)** Encontrar a melhor correlação / similaridade / distância. **iii**) Avaliar as recomendações geradas. Em cada item previsto para os experimentos, foi utilizado o conjunto apropriado de dados bem como as métricas / métodos mais utilizadas para realizar o experimento.

5. Experimentos e Resultados

Esta seção apresenta os experimentos e resultados das três etapas da abordagem proposta anteriormente.

5.1. Pesos Específicos para o Rep-Index

A determinação dos pesos específicos do Rep-Index para cada área de estudo foi realizada anteriormente utilizando-se o complemento para o Rep-Index proposto por [Vivian et al. 2016]. O mesmo utiliza técnicas de mineração de dados e aprendizado de máquina para computar cinco opções de pesos. A avaliação da melhor opção é realizada pela correlação de Spearman. Dessa forma, o Rep-Index é adequado para classificar os pesquisadores o mais próximo possível da realidade da área (pesquisadores do CNPq).

5.2. Similaridade entre as Subáreas

Para localizar a melhor combinação de técnicas, deve-se inicialmente agrupar os pesquisadores em categorias (subáreas do currículo Lattes). Utilizou-se o elemento Área de Atuação (AA) do Rep-Model modificado para esta finalidade. Em um grupo formado por 398 pesquisadores da área de Ciência da Computação, inicialmente foram localizadas 959 subáreas de atuação. Essa quantidade se justifica no fato de que a maioria dos pesquisadores apresenta mais de uma subárea de atuação, em geral uma área clássica da Ciência da Computação e algumas áreas de pesquisas mais atuais ou mesmo multidisciplinares. Ao final foram obtidas 219 categorias distintas e mais uma categoria denominada *Empty* para os casos sem o elemento AA. Entre as 220 categorias, 57 (25,90%) possuem mais de um pesquisador e 163 (74,09%) são formadas por apenas um único pesquisador. As classes com apenas um pesquisador foram retiradas dos experimentos.

A partir das matrizes de similaridades (uma para cada função) entre os pesquisadores e as categorias, aplica-se o algoritmo do vizinho mais próximo (*Nearest Neighbor*)

[1] http://lattes.cnpq.br
[2] https://scholar.google.com.br
[3] http://basex.org
[4] https://github.com/grvivian/xml2arff

com o parâmetro n (indica quantos vizinhos devem ser selecionados) igual ao número de pesquisadores existentes na categoria em questão. Dessa forma, seleciona-se os n pesquisadores mais afins à categoria. Isto é obtido pela ordenação decrescente das similaridades do pesquisador, seguido pela seleção dos n primeiros pesquisadores. Após localizar os pesquisadores mais afins para cada categoria, basta comparar com a definição original das categorias e encontrar os verdadeiros positivos (TP), falsos positivos (FP), verdadeiro negativo (TN) e falso negativo (FN). Com essas informações, constrói-se a matriz de confusão e aplica-se as métricas de avaliação.

Os experimentos foram realizados no Apache Mahout[5] versão 1.12.2. O ambiente já possui classes Java prontas para diversos idiomas, cada uma com as regras pré determinadas de análise léxica, *stemming* e *stopwords*. Devido ao fato de que as informações textuais estarem escritas principalmente nos idiomas Português e Inglês, optou-se por realizar experimentos com ambas as classes. Além das existentes, criou-se uma nova classe em Java denominada `MyBrazilianAnalyzer.java`, a qual possui regras personalizadas de *stopwords* e sinonímia. Também criou-se a classe `LoglikelihoodDistanceMeasure.java` para computar esta medida de distância com vetores esparsos, uma vez que o Mahout possui o `Log-Likelihood` apenas para filtragem colaborativa. As palavras com frequência relativa (TF) abaixo de 2 foram desconsideradas. Na Tabela 2 pode-se visualizar as classes utilizadas.

Tabela 2. Classes empregadas nos experimentos.

Classe	Análise léxica	*Stopwords*	Sinonímia	*Stemming*	Dicionário
ClassicAnalyzer (CL)	ClassicTokenizer, ClassicFilter, LowerCaseFilter	33 (Inglês)	Não	Não	44.723
StandardAnalyzer (ST)	StandardTokenizer, StandardFilter, LowerCaseFilter	33 (Inglês)	Não	Não	44.360
EnglishAnalyzer (EN)	StandardTokenizer, StandardFilter, LowerCaseFilter	33 (Inglês)	Não	PorterStemFilter	35.379
BrazilianAnalyzer (BR)	StandardTokenizer, StandardFilter, LowerCaseFilter	128 (Português)	Não	BrazilianStemFilter	33.290
MyBrazilianAnalyzer (MY)	StandardTokenizer, StandardFilter, LowerCaseFilter	1234 (Português)	147*	BrazilianStemFilter	32.842

*Usou-se a Sinonímia para Traduzir termos da área em língua Inglesa para língua Portuguesa.

Observa-se que a classe `ClassicAnalyzer` foi a que gerou o maior dicionário de palavras. Ficando em segundo lugar a classe `StandardAnalyzer`, em seguida `EnglishAnalyzer`, `BrazilianAnalyzer` e `MyBrazilianAnalyzer`.

A partir das similaridades apresentadas e das classes da Tabela 2 realizou-se as avaliações com as métricas: *F-Measure* (média harmônica entre *precision* e *recall*), RMSE e *Kappa*. Utilizou-se a configuração *full-training set* para avaliar o modelo. Os valores foram obtidos através da média ponderada entre o resultado de cada classe e o seu número de pesquisadores. Na Figura 2 pode-se visualizar as métricas acima mencionadas.

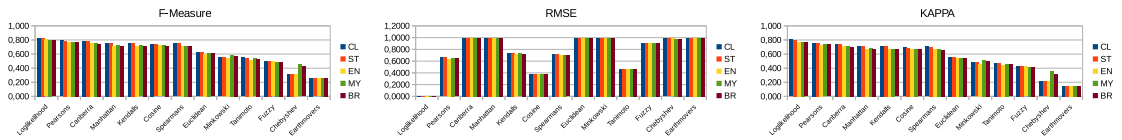

Figura 2. Métricas *F-Measure*, RMSE e *Kappa*.

Observa-se que a classe `ClassicAnalyzer` (maior dicionário) em conjunto com a técnica `Log-Likelihood` foi a que obteve a maior *F-Measure* (0,831). Além disso, esta combinação apresentou os menores erros para RMSE (0,0012). As correlações

[5]http://mahout.apache.org

de Pearson e Kendall, similaridade Fuzzy, distância do Cosseno e Tanimoto apresentam RMSE entre 0,4 e 0,85. No entanto, as demais funções, por apresentarem domínio positivo infinito acabaram ficando com os valores muito próximos de zero quando convertidas em similaridades pela equação mencionada na abordagem. Dessa forma, o RMSE delas está muito próximo de 1,0. O resultado da métrica estatística Cohen's *Kappa* (0,806) também obteve a melhor colocação. A medida estatística *Kappa* está no intervalo de 0,80 até 0,90 (resultado forte). Com relação as demais classes, verifica-se que as mesmas apresentam resultados ligeiramente inferiores a `ClassicAnalyzer`. Um fato também observado é a enorme diferença nos erros (RMSE) entre a técnica `Log-Likelihood` e as demais. Isto indica um alto grau de precisão nas similaridades, ou seja, valores muito próximos de 1,0 devido ao fato da mesma ser uma função monótona crescente.

5.3. Recomendações

Para realizar os experimentos com as recomendações, utilizou-se o grupo total de 398 pesquisadores do CNPq, os grupos individuais de bolsa 1A (23), 1B (22), 1C (38), 1D (50) e 2 (264); e mais um grupo de teste com 143 pesquisadores composto por 80 docentes do grupo INF da UFRGS[6] e 63 docentes do DCC da UFMG[7]. Destes 64 são bolsistas de produtividade do CNPq e 79 não possuem bolsas do CNPq. Solicitou-se a geração de recomendações para rede de colaboradores e para Grau de Instrução. Na Figura 3 pode-se visualizar a métrica *coverage* adaptada à abordagem com o parâmetro n variando de 1 até 50 recomendações. Não se utilizou a *recall* e *precision* pois só podem ser empregadas em situações onde pode-se predizer se o usuário gostou ou não da recomendação.

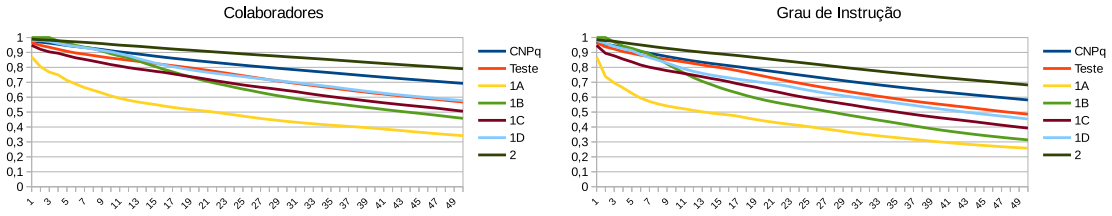

Figura 3. *Coverage* de Recomendações para Colaboradores e Grau de Instrução.

Nas Figuras anteriores pode-se observar que nunca ocorreu a situação onde não existe o que recomendar (valores zerados). Também fica evidente que os grupos iniciais do CNPq, ou seja, os grupos 2 e 1D possuem valores maiores de *coverage* do que os grupos mais avançados (1C, 1B e 1A). Isto se justifica pelo fato que os grupos finais têm maior reputação no Rep-Index e Grau de Instrução do que os grupos iniciais e de teste. As recomendações geradas estão disponíveis em um repositório[8] como material suplementar. Os pesquisadores estão identificados apenas pela id da plataforma Lattes.

Ao final, gerou-se o conjunto total das 28 possíveis recomendações diferentes (personalizadas e não personalizadas), uma para cada elemento do tipo inteiro do Rep-Model. Neste experimento foi utilizado o limiar de 0,99905 para limitar a similaridade entre os pesquisadores. As recomendações que não incrementam a reputação (Rep-Index) do pesquisador foram desconsideradas, isto significa que os elementos que não tiveram

[6]http://www.inf.ufrgs.br/site/pessoas/corpo-docente/
[7]http://www.dcc.ufmg.br/dcc/?q=pt-br/professores
[8]https://github.com/grvivian/ERBD2017/

valores para os pesos do Rep-Index foram ignorados e portanto não serão recomendados. Foi computada a média da métrica *coverage* e da *diversity* com n variando de 1 até 20. A primeira indica a capacidade da abordagem em gerar recomendações e a última avalia a diversidade apenas do conjunto de itens recomendados. A Figura 4 apresenta as mesmas.

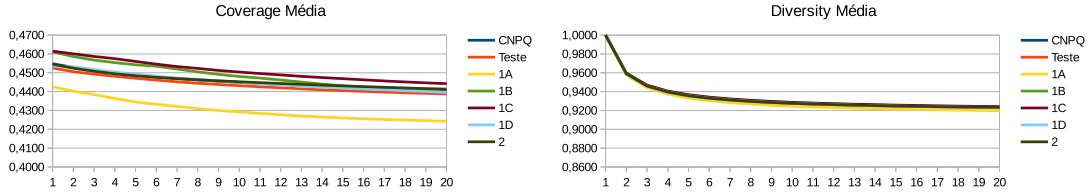

Figura 4. *Coverage* Média e *Diversity* Média para Ciência da Computação.

Observa-se um comportamento semelhante para a *coverage* da Figura 3 porem com valores menores. Com relação a *diversity*, pode-se constatar que quando $n = 1$ ocorre o máximo valor, a partir de $n \geq 2$ a mesma decresce para 0,92 quando $n = 20$. Em todos as situações a diversidade de elementos recomendados foi satisfatória. Na Tabela 3 pode-se visualizar as recomendações geradas para um pesquisador do nível SR.

Tabela 3. Recomendações geradas.

Recomendação	Tipo	Peso	Máx.	Inc.	Aumento
Aumente o item: Orientação de Doutorado (PTA) para 6	REC_PTA	16,789	46	1	0,365
Aumente o item: Orientação de Pós-doutorado (PA) para 1	REC_PA	5,273	19	1	0,278
Aumente o item: Membro de Corpo Editorial de Periódico (EBM) para 6	REC_EBM	4,569	18	1	0,254
Aumente o item: Livro (BP) para 7	REC_BP	5,187	57	1	0,091
Aumente o item: Orientação de Mestrado (MDA) para 18	REC_MDA	9,328	114	1	0,082
Aumente o item: H-Index (HI) para 18	REC_HI	8,609	116	1	0,074
Aumente o item: Artigo em Periódico (ASJ) para 24	REC_ASJ	17,420	246	1	0,071
Aumente o item: Revisão de Periódico (RJ) para 1	REC_RJ	5,146	77	1	0,067
Aumente o item: Participação em Banca de Mestrado (PEBPT) para 56	REC_PEBPT	8,434	127	1	0,066
Aumente o item: Prêmios (PREM) para 6	REC_PREM	3,812	59	1	0,065
Aumente o item: Capítulo de Livro (BCP) para 12	REC_BCP	3,644	62	1	0,059
Amplie a sua Rede de colaboração com: 5554254760869075, similaridade: 0,999155 Rep-Index: 29,65	REC_NC	4,387	158	1	0,028
Aumente o item: Trabalho Completo em Conferência (CWPCP) para 71	REC_CWPCP	7,401	470	1	0,016

Na tabela anterior observa-se que as recomendações estão relatadas em ordem de relevância para a reputação, isto responde ao questionamento de “quando fazer”. As próprias recomendações respondem ao questionamento ”do que fazer” e as personalizações respondem ”como fazer”.

6. Conclusões e Trabalhos Futuros

Este trabalho apresentou uma abordagem para gerar recomendações de plano de carreira de pesquisadores utilizando a similaridade de perfil e reputação acadêmica. A similaridade de perfil foi definida com base nas informações textuais do Rep-Model por meio da técnica de TF-IDF. Foi realizado um experimento com o objetivo de localizar a melhor técnica para esta tarefa. Criou-se para isso, diversas categorias que possuem todas as informações textuais de seus pesquisadores e comparou-se as mesmas com o conjunto total de pesquisadores. Experimentou-se três funções de correlação, nove de distância e uma de similaridade, bem como cinco classes de pré-processamento. Ao final, observou-se que a combinação `Log-likelihood` e `ClassicAnalyzer` obteve os melhores resultados. Ao final dos experimentos, gerou-se as recomendações personalizadas e não personalizadas para diversos grupos de teste. As recomendações foram avaliadas com base na métrica *coverage* e *diversity* e apresentaram resultados satisfatórios. Por se tratar de uma abordagem inédita não temos referências para comparar com o *baseline*.

Referências

Beel, J., Langer, S., Genzmehr, M., Gipp, B., Breitinger, C., e Nürnberger, A. (2013). Research paper recommender system evaluation: A Quantitative Literature Survey. In *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation - RepSys '13*, number April, páginas 15–22, New York, New York, USA. ACM Press.

Cervi, C. R., Galante, R., e 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., e 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.

Ekstrand, M. D., Kannan, P., Stemper, J. A., Butler, J. T., Konstan, J. A., e Riedl, J. T. (2010). Automatically Building Research Reading Lists. *RecSys2010*, páginas 159–166.

Huang, W., Kataria, S., Caragea, C., Mitra, P., Giles, C. L., e Rokach, L. (2012). Recommending citations. *Proceedings of the 21st ACM international conference on Information and knowledge management - CIKM '12*, página 1910.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632.

Middleton, S. E., Shadbolt, N. R., e De Roure, D. C. (2004). Ontological user profiling in recommender systems. *ACM Transactions on Information Systems*, 22(1):54–88.

Page, L., Brin, S., Motwani, R., e Winograd, T. (1999). The pagerank citation ranking: bringing order to the web. *Stanford InfoLab*.

Sugiyama, K. e Kan, M.-Y. (2013). Exploiting Potential Citation Papers in Scholarly Paper Recommendation. *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries - JCDL '13*, página 153.

Sugiyama, K. e Kan, M. Y. (2015). A comprehensive evaluation of scholarly paper recommendation using potential citation papers. *International Journal on Digital Libraries*, 16(2):91–109.

Vivian, G. R. e Cervi, C. R. (2016a). Utilizando técnicas de data science para definir o perfil do pesquisador brasileiro da area de ciência da computação. *XII Escola Regional de Informática de Banco de Dados*, páginas 108–117.

Vivian, G. R. e Cervi, C. R. (2016b). xml2arff: Uma ferramenta automatizada de extração de dados em arquivos xml para data science com weka e r. *XII Escola Regional de Informática de Banco de Dados*, páginas 159–162.

Vivian, G. R., Cervi, C. R., e Rovadosky, D. N. (2016). Using selection attribute algorithms from data mining to complement the rep-index. *IADIS International Journal on WWW/Internet*, 15:219–226.

Zhang, Z. e Li, L. (2010). A research paper recommender system based on spreading activation model. In *The 2nd International Conference on Information Science and Engineering*, páginas 928–931. IEEE.

Um *Survey* sobre Extração de Esquemas de Documentos JSON

Rudimar Imhof[1], Angelo Augusto Frozza[1,2], Ronaldo dos Santos Mello[1]

[1]Departamento de Informática e Estatística (INE) – Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88.049-900 – Florianópolis – SC – Brasil

[2]Instituto Federal Catarinense (IFC) – Campus Camboriú
Caixa Postal 2016 – 88.340-055 – Camboriú – SC – Brasil

rudimar.imhof@gmail.com, frozza@ifc-camboriu.edu.br, r.mello@ufsc.br

Abstract. *JSON (JavaScript Object Notation) is a format for representation and interchange of complex and heterogeneous data that is becoming very popular. In this context, initiatives related to the integration or querying of large volumes of JSON data must deal with the problem of defining an unified schema for a set of relevant JSON documents. This paper aims to present a review on schema extraction from document in JSON format. The contributions of the paper are the presentation of approaches related to the theme as well as a comparative analysis of them.*

Resumo. *JSON (JavaScript Object Notation) é um formato de representação e intercâmbio de dados complexos e heterogêneos que vem crescendo em popularidade. Nesse sentido, iniciativas no sentido de integrar ou consultar grandes volumes de dados neste formato se deparam com a problemática de definir um esquema unificado para um conjunto de documentos JSON de interesse. Este artigo tem por objetivo apresentar um levantamento sobre extração de esquemas de documentos em formato JSON. As contribuições deste artigo são a apresentação de abordagens relacionados ao tema e uma análise comparativa destas abordagens.*

1. Introdução

Diversos autores têm manifestado que o movimento de bancos de dados voltados ao gerenciamento de dados complexos e semiestruturados tem crescido vertiginosamente. Como consequência deste crescimento, estima-se que boa parte dos dados, hoje no mundo, encontram-se armazenados em bancos de dados desse tipo [3]. Como exemplo, a utilização do Sistema Gerenciador de Banco de Dados (SGBD) *MongoDB*, um banco de dados (BD) NoSQL orientado a documentos, já superou a utilização do banco de dados relacional *PostgreSQL*, ficando atrás apenas de *Oracle*, *MySQL* e *Microsoft* SQL *Server* [1]. Entre as principais características desses bancos de dados estão a capacidade de representar dados complexos, a escalabilidade para gerenciar tanto grandes conjuntos de dados quanto o aumento do tráfego de dados, e a falta de esquemas ou o uso de esquemas flexíveis [12].

Nesse contexto encontra-se o formato de dados JSON (*JavaScript Object Notation*), que vem sendo utilizado cada vez mais para a representação e o intercâmbio de dados complexos e heterogêneos em diversos domínios de aplicação. Devido à

crescente utilização de bancos de dados NoSQL ou outros tipos de repositórios que suportam esse formato, surge a necessidade de integração ou acesso integrado a tais repositórios de dados.

Um problema associado a essas necessidades é a extração e unificação de esquemas de coleções de dados no formato JSON visando facilitar a futura manipulação desses dados. Assim sendo, este artigo apresenta um *survey* que descreve sucintamente e analisa trabalhos que visam a extração de esquemas de dados presentes para repositórios de dados no formato JSON. De forma complementar, é apresentado um quadro comparativo desses trabalhos e uma análise dos mesmos com base neste quadro. Além do problema de integração de dados já citado, esquemas extraídos de bancos de dados NoSQL podem ser úteis para a aquisição de conhecimento no desenvolvimento de sistemas, na reengenharia de sistemas, em processos de migração de sistemas e conversão de dados, entre outros [4].

Este artigo está organizado da seguinte forma: a seção 2 apresenta brevemente o formato de dados JSON. A seção 3 apresenta os trabalhos relacionados e suas principais características. A seção 4 apresenta uma análise comparativa dos trabalhos analisados. A seção 5 finaliza o estudo e apresenta as considerações finais.

2. O Formato de Dados JSON

JSON é um formato leve de intercâmbio de dados baseado na linguagem *JavaScript* [10]. Sua ascensão ocorreu devido à facilidade de leitura e escrita neste formato, tanto por humanos como por máquinas. O conteúdo de um documento JSON encontra-se em formato de texto, independente de linguagem de programação, mas que usa convenções que são familiares às usadas em linguagens como C, C++, *Java*, *Perl*, *Python*, entre outras. JSON é construído sobre duas estruturas compatíveis com estruturas existentes nessas linguagens de programação:

- Um conjunto de pares chave-valor, semelhante a um objeto ou registro;
- Uma lista ordenada de valores tratada como um *array*, vetor, lista ou sequência.

Figura 1 – Exemplo de documento no formato JSON

```
{
       "id": "00000234567894",
       "name": "Jane Doe",
       "birthday": "04/18/1978",
       "gender": "female",
       "type": "user",
       "work": [{
          "employer": {
             "id": "106119876543210",
             "name": "Doe Inc."
          },
          "start_date": "2007 - 08"
        },
        {
          "start_date": "2004",
          "end_date": "2007"
       }]
}
```

(Fonte: adaptado de [7])

A terminologia que define a estrutura de um documento JSON compreende os seguintes conceitos: (i) *Objeto*: um conjunto não ordenado de pares chave-valor. Um

objeto inicia com '{' e termina com '}', cada chave é seguida por ':' e os pares chave-valor são separados por ','; (ii) *Array*: uma coleção ordenada de valores que inicia com '[' e termina com ']', sendo cada elemento (valor) do *array* separado por ','; (iii) *Valor*: pode ser um tipo primitivo (*string*, número ou *booleano*), um valor nulo, um objeto ou um *array*. O uso de objetos e *arrays* como valores permite definir estruturas aninhadas.

A Figura 1 apresenta um exemplo de documento no formato JSON que descreve o perfil (*objeto*) de um usuário de rede social. As chaves *id*, *name*, *birthday*, *gender*, *type*, *start_date* e *end_date* possuem valores com tipos primitivos; o valor da chave *work* é um *array* com dois elementos do tipo *objeto*; o valor da chave *employer* também é um *objeto*.

3. Trabalhos Relacionados

Esta seção apresenta sucintamente, devido às limitações de espaço, as principais características dos trabalhos existentes na literatura que lidam com extração de esquemas de dados JSON. Para a seleção dos trabalhos, foi feita uma busca em sete bases de dados acadêmicas (*Science Direct*[1], ACM DL[2], IEEE *Xplore*[3], DBLP[4], Portal Periódicos CAPES[5], *Scopus*[6], *Web of Science*[7]) pelos termos "NoSQL", "*Reverse Engineering*", "JSON" e "NoSQL *Schema*". A busca limitou-se apenas a artigos em Inglês, publicados em Conferências e *Journals*, disponíveis publicamente. Após análise dos artigos, foram selecionados apenas os que apresentavam contribuições para o tema "engenharia reversa de bases de dados NoSQL".

O trabalho de Kapsammer *et al*. [7] tem por objetivo a extração e transformação de esquemas de perfis de usuário em redes sociais para a criação de perfis integrados. Para isso, propõe um processo em 4 etapas: (*i*) extração de instâncias de dados JSON - é obtido um conjunto de múltiplas amostras de dados diferentes (por meio de requisições às APIs – *Application Programming Interface* - de redes sociais), cada uma possuindo, eventualmente, fragmentos de dados complementares; (*ii*) extração de esquemas JSON das instâncias – múltiplos esquemas são extraídos dos registros obtidos na fase anterior. Esses esquemas são usados para conceber um único esquema consistente através de uma operação de *merge*; (*iii*) transformação dos esquemas JSON para esquemas no metamodelo ECORE [13] – um modelo canônico que possibilita a transformação de esquemas para outros formatos, como XML *Schema*, OWL (*Web Ontology Language*), de classe *Java* etc.; (*iv*) criação de perfis de usuário integrados – são aplicados processos de integração com o suporte de ferramentas de modelagem de propósito geral (como EA), ferramentas para verificação de similaridade (como COMA++) e para mapeamento de esquemas (como *MapForce*). Na etapa de extração de esquemas, o conhecimento prévio sobre o conjunto de esquemas pode ser utilizado para configurar estratégias de

[1] http://www.sciencedirect.com/
[2] http://dl.acm.org/ - opção "*The ACM Guide to Computing Literature*"
[3] http://ieeexplore.ieee.org/Xplore/home.jsp
[4] http://dblp.uni-trier.de/
[5] www.periodicos.capes.gov.br
[6] https://www.scopus.com/
[7] https://webofknowledge.com/

generalização e fusão. Esse conhecimento pode ser recuperado da documentação das APIs, das convenções de nomenclatura ou da investigação manual de exemplos.

O trabalho de Izquierdo & Carbot [5] apresenta uma abordagem para gerar o esquema básico de um conjunto de documentos JSON obtidos de APIs de serviços *Web*. O processo é composto por 3 etapas dirigidas por *Model Driven Engineering* (MDE): (*i*) pré-descoberta - extrai modelos JSON de baixo nível (usando *XText*[8], um *framework open source* para definir linguagens de programação, para definir um metamodelo JSON); (*ii*) descoberta de serviço único - visa a obtenção de informações de esquema para um dado serviço; (*iii*) descoberta multiserviço - encarregado de compor as informações de esquema obtidas na fase anterior, a fim de obter uma visão geral do domínio da aplicação. Na etapa *(i)*, documentos JSON são obtidos a partir de múltiplas chamadas a serviços de uma API e são transformados para um metamodelo JSON. Na etapa *(ii)*, o processo é executado para cada objeto JSON em dois modos de execução: criação de um novo esquema ou refinamento de um esquema existente. Seu produto é um modelo de domínio para cada serviço. Na etapa *(iii)* é realizado o *merge* dos modelos de domínio de serviço obtidos na etapa *(ii)*, produzindo o modelo de domínio da aplicação.

O trabalho de Kiran & Vijayakumar [8] propõe um sistema de integração semântico baseado em ontologia para bancos de dados NoSQL orientados a colunas, como o *HBase*. A abordagem de integração se dá por meio de um *endpoint* RDF sobre o qual se pode formular consultas usando ontologias combinadas obtidas de bancos de dados distintos. Apesar de seguir estratégias maduras para integração de bancos de dados relacionais, os autores destacam que há diferenças de implementação quando se trabalha com bancos de dados NoSQL. Considerando apenas o módulo de extração de esquemas, o trabalho propõe duas alternativas: a) geração de esquemas *online*, para bancos de dados sendo populados; b) geração de esquemas *offline*, para bancos de dados já populados. A extração de esquemas em cada banco de dados considera as seguintes atividades: (*i*) através de um *job MapReduce*, criar uma tabela de consulta no *HBase* para cada tabela que terá o esquema extraído; (*ii*) extrair o esquema com o suporte de um algoritmo genético que busca o esquema mais apto (a aptidão é definida como o número total de colunas do indivíduo e que possivelmente representa os demais indivíduos do banco de dados); (*iii*) criar um *HashMap in-memory* com os detalhes da família de coluna do esquema mais apto. O esquema obtido é então mapeado para uma ontologia OWL (esquema local). Uma vez que a proposta é de um sistema de integração de bancos de dados, em qualquer momento no sistema, dois esquemas existem: *a*) um esquema local para cada cliente; *b*) um esquema global acessível a todos os clientes.

O trabalho de Klettke *et al*. [9] propõe um algoritmo para extração de esquemas e produção de medidas de similaridade que capturam o grau de heterogeneidade de um conjunto de documentos JSON, além de revelar discrepâncias estruturais nos dados. O algoritmo é executado em 4 etapas: (*i*) seleção de documentos - executa a extração de esquemas sobre a coleção completa ou sobre um subconjunto de documentos JSON selecionados; (*ii*) extração da estrutura dos documentos JSON; (*iii*) construção do *Structure Identification Graph* (SG) – para cada conceito do documento JSON um nodo na árvore SG é adicionado ou estendido e uma aresta ligando ao nodo pai é adicionada ou estendida; (*iv*) criação do esquema JSON. Adicionalmente, os dados no grafo SG são

[8] http://www.eclipse.org/xtext

usados para produzir estatísticas, encontrar discrepâncias estruturais nos dados e calcular medidas que identificam quão regular são os documentos da coleção (grau de cobertura). O grau de cobertura é um valor entre 0 e 1 que especifica a sobreposição entre dois ou mais documentos, ou seja, quanto os documentos são similares. Discrepâncias estruturais permitem identificar estruturas que ocorrem raramente ou erros reais na estrutura de dados JSON.

O trabalho de Ruiz *et al*. [12] propõe um processo de engenharia reversa para bancos de dados NoSQL orientados a documentos através da metodologia MDE (*Model Driven Engineering*) [13]. O processo é realizado em 3 etapas: (*i*) extração de objetos JSON – através de um *job MapReduce* que extrai uma coleção de objetos JSON contendo um objeto para cada versão de uma entidade; (*ii*) definição do esquema JSON – cada objeto JSON é injetado em um modelo que está em conformidade com um metamodelo JSON proposto pelos autores. Isto é feito mapeando os elementos da gramática JSON em elementos do metamodelo; (*iii*) definição do esquema NoSQL – o processo de engenharia reversa é implementado como uma transformação *model-to-model* sobre cada modelo JSON, gerando modelos de domínio (com diferentes versões) em conformidade com um metamodelo de esquema NoSQL. Os modelos de esquema NoSQL inferidos podem ser usados para construir utilitários de bancos de dados, que exigem conhecimento da estrutura do BD, como, por exemplo, SQL *query engines*, e, ferramentas auxiliares, como validadores de dados, *scripts* de migração ou diagramas de esquemas.

O trabalho de Wang [14] apresenta um *framework* de gerenciamento de esquemas para bancos de dados NoSQL orientados a documentos. O *framework* descobre e persiste esquemas de documentos JSON e também suporta consultas sobre os esquemas e sumarização de esquemas. O trabalho propõe uma nova estrutura de dados, chamada *eSiBu-Tree*, para armazenar e dar suporte a consultas a esquemas, bem como o conceito de *skeleton* para visualização de esquemas, o qual representa o menor conjunto de atributos que melhor definem o núcleo do esquema. Documentos JSON (chamados registros JSON) são obtidos a partir de um *Dataset* JSON e, para cada registro, um *record schema* é extraído por meio da representação da estrutura do documento na *eSiBu-Tree*. Apenas os *labels* dos campos são considerados para montar a estrutura do registro na *eSiBu-Tree* (não fazendo referência a tipos de dados). O *skelenton* é usado para gerar um esquema único (*core schema*) do objeto, o qual é formado pelos atributos que aparecem com maior frequência nos esquemas dos registros.

O trabalho de Discala & Abadi [2] propõe um algoritmo que automaticamente transforma dados aninhados e desnormalizados, comumente encontrados em bancos de dados NoSQL, em dados relacionais que podem ser armazenados em um banco de dados convencionais. O algoritmo descobre dependências funcionais entre os atributos a fim de organizar esses atributos em tabelas relacionais. O processo apresenta 3 etapas: (*i*) criação de uma árvore de atributos e mineração de dependências funcionais entre atributos - as dependências são usadas para identificar grupos de atributos que podem corresponder a uma entidade independente que, posteriormente, dá origem a uma tabela; (*ii*) identificação de entidades de domínio sobrepostas – pesquisa a árvore de atributos para descobrir entidades semanticamente equivalentes espalhadas no conjunto de dados; (*iii*) agrupamento dos resultados intermediários para produzir o esquema físico relacional, mesmo que não esteja em conformidade com as tradicionais formas normais. Uma característica desse trabalho é que ele não faz uso da estrutura previamente existente em um documento JSON, buscando reconstruir a estrutura apenas pela identificação de

dependências funcionais presentes nos dados.

O trabalho de Liu *et al*. [11] apresenta o JSON *DataGuide*, que é um esquema auto computado, dinâmico e flexível, para coleções de documentos JSON, implementado no *Oracle 12cR2 release*. Uma coleção de documentos JSON é armazenada em uma coluna com a restrição "*IS JSON*". Para cada coluna JSON é também armazenado um JSON *DataGuide*, que é incrementalmente atualizado conforme novos documentos são inseridos. O *Oracle* não usa os esquemas gerados para o armazenamento de dados JSON, mas sim, para permitir consultas sobre uma visão relacional dos dados JSON usando SQL/JSON, uma linguagem de consulta baseada em caminhos (*paths*) JSON DOM. Com isso, introduz-se um paradigma *"escreva sem esquema, leia com esquema"*. O JSON *DataGuide* mantém uma derivação de todos os *paths* estruturais hierárquicos existentes em uma coleção JSON, os tipos de dados e as estatísticas dos valores escalares das folhas. Um JSON *DataGuide* para uma única instância de documento JSON é obtido pela extração do "esqueleto" dos nós *containers* (objetos e *arrays*) da árvore JSON DOM. Valores escalares nas folhas são trocados pelo tipo e comprimento dos dados. O JSON *DataGuide* para uma coleção de documentos é obtido pela combinação (*merge*) das instâncias *DataGuide* dos documentos da coleção, removendo os *paths* da árvore duplicados e que tem o mesmo tipo de nó de árvore. Caminhos com tipo de nó de árvore diferente são considerados diferentes. A informação de dados escalares folha é combinada, eliminando conflitos de tipos de dados por meio da definição de um tipo mais geral e usando o maior tamanho. Adicionalmente, o *DataGuide* armazena informações estatísticas dos caminhos, tais como frequência, valores mínimos e máximos e número de valores nulos. Por fim, o *Oracle* fornece um conjunto de procedimentos PL/SQL que permitem projetar visões relacionais e colunas virtuais dos dados JSON a partir do JSON *DataGuide*. Observa-se que o JSON *DataGuide* é aditivo, isto é, ele não remove *paths* quando documentos JSON são excluídos. Funções SQL para calcular o JSON *DataGuide* dinamicamente sobre os resultados de qualquer consulta SQL que retorne um conjunto de documentos JSON também são disponibilizadas.

4. Análise Comparativa

O Quadro 1 apresenta um comparativo das principais características observadas nos trabalhos apresentados na seção anterior. Além do identificador da referência do trabalho, as características consideradas são as seguintes: (*i*) *Origem* (fonte de origem usada para obter documentos JSON); (*ii*) *Objetivo* (perspectiva de uso dos esquemas extraídos); (*iii*) *Abordagem* (abordagem adotada para obter o esquema final); (*iv*) *Modelo Intermediário* (se o processo usa algum modelo de dados intermediário); (*v*) *Modelo de Saída* (como o esquema é disponibilizado ao final do processo); (*vi*) *Unificação* (se existe um processo de unificação de vários esquemas JSON visando obter um esquema único); e (*vii*) *Etapas do Processo* (um resumo das etapas do processo de extração adotado).

Com relação à *fonte de origem* dos documentos JSON, percebe-se que os trabalhos se concentram em três categorias: *a*) dados provenientes de APIs de serviços *Web*; *b*) *Datasets* de documentos JSON; *c*) Bancos de dados NoSQL. As duas últimas categorias podem ser consideradas como do mesmo tipo de fonte, uma vez que representam coleções de documentos JSON. Uma característica das APIs é que dois ou mais objetos no mesmo ou em diferentes documentos JSON gerados por uma chamada ao mesmo serviço não necessariamente têm a mesma estrutura exata, ou seja, é possível que alguns documentos possuam somente um subconjunto dos metadados por causa de parâmetros e filtros pas-

Quadro 1: Comparativo dos trabalhos analisados

Id	Origem	Objetivo	Abordagem	Modelo Interm.	Modelo de Saída	Unif.	Etapas do processo
[7]	APIs de redes sociais	Integrar perfis de usuário em redes sociais	Transformação de modelos	JSON *Schema*	Modelo de classes (ECORE)	SIM	a) Extração de dados b) Extração de esquemas c) Transformação d) Integração
[5]	APIs de serviços *web*	Criar visão de domínio para os serviços da API	Transformação de modelos	Meta-modelo JSON (ECORE)	Modelo de domínio da Aplicação (ECORE)	SIM	a) Pré-descoberta de documentos JSON b) Extração de esquema do serviço c) Criação do esquema de domínio
[8]	*HBase*	Integrar bancos de dados	Organização em ontologia	*HBase*	OWL	SIM	a) Criação de uma tabela de consulta sobre os documentos JSON com *MapReduce* b) Escolha do esquema mais apto via algoritmo genético c) Mapeamento do esquema para uma ontologia local OWL d) Combinação de ontologias para formar uma ontologia global (integração)
[9]	*Dataset* no *MongoDB*	Criar ferramentas para manipular e gerenciar esquemas	Organização hierárquica (grafo)	*Structure Identification Graph* (SG)	JSON *Schema*	SIM	a) Seleção de documentos JSON b) Extração estrutura dos documentos c) Construção do grafo SG d) Geração do esquema JSON
[12]	*MongoDB*, *CouchDB* e *HBase*	Criar utilitários para BD NoSQL	Transformação de modelos	Meta-modelo JSON	Meta-modelo de esquema NoSQL	NÃO	a) Extração de objetos JSON b) Transformação para esquemas JSON c) Transformação para esquema NoSQL
[14]	*Datasets* JSON	Consultar esquemas e integrar dados	Organização Hierárquica (árvore)	*eSiBu-Tree*	*eSiBu-Tree*	SIM*	a) Seleção dos documentos JSON b) Criação registro na *eSiBu-Tree* c) Visualização do esquema unificado (*skeleton*)
[2]	*Dataset* JSON ou CSV	Migrar para BD Relacional	Organização Hierárquica (grafo)	Grafo dirigido	Modelo relacional	SIM	a) Criação de uma árvore de atributos e mineração de dependências funcionais b) Identificação de entidades sobrepostas c) Geração do esquema físico relacional
[11]	*Oracle* JSON *column*	Consultar coleções de documentos JSON	Organização hierárquica (JSON DOM)	--x--	JSON *DataGuide*	SIM	a) Derivação dos caminhos (*paths*) estruturais hierárquicos de documentos JSON b) Armazenamento de informações estatísticas dos JSON *paths*

sados ao serviço, reduzindo a quantidade de pares chave-valor (por exemplo, para diminuir o tráfego de rede) [5]. Documentos JSON retornados de *Datasets* ou bancos de dados NoSQL, por sua vez, podem conter toda a informação do esquema em uma única instância.

Quanto ao *objetivo*, percebe-se que a maioria dos trabalhos visam a criação de ferramentas para manipular e gerenciar esquemas [5,7,9,12,14]. Dois trabalhos enfatizam o uso de dados JSON em BDs relacionais [2,11] e apenas 2 trabalhos mencionam o uso dos esquemas para integração de fontes de dados distintas [8,14].

Em relação a *abordagem* utilizada no processo de extração, são identificadas três categorias: *a*) organização hierárquica [2,9,11,14], que utiliza alguma estrutura em árvore ou grafo para suportar o processo de extração ou representação de esquemas; b) transformação de modelos [5,7,12], que faz uso de técnicas de *Model Driven Engineering* (MDE) [13] para definir um esquema para os dados JSON. Gera-se um esquema para cada documento JSON e, posteriormente, faz-se a unificação dos esquemas em um único esquema de domínio; *c*) organização baseada em ontologia [8], que aposta em tecnologias da *Web* semântica para realizar a integração de dados.

Pode-se fazer uma correlação entre o objetivo do esquema e a abordagem adotada no processo: propostas visando a criação de ferramentas para trabalhar com esquemas JSON adotam, preferencialmente, alguma técnica de transformação de modelos. Já propostas que visam a integração de dados ou a realização de consultas a um esquema geralmente adotam alguma estrutura hierárquica ou ontologia para representar o esquema final.

A grande maioria dos trabalhos utiliza um modelo de dados intermediário diferente do modelo de dados usado para representar o esquema JSON final. Ele é utilizado geralmente nas etapas iniciais do processo com o objetivo de representar a estrutura de uma única instância de documento JSON. Alguns trabalhos propõem uma estrutura própria, em formato de árvore ou grafo, utilizada para representar a estrutura de objetos aninhados de documentos JSON [2,9,14]. Abordagens baseadas em MDE [5,12] usam metamodelos ECORE [13], tirando vantagem das ferramentas providas por essa tecnologia, por exemplo, para representar metamodelos como diagramas de classe UML. Trabalhos que citam o uso de esquemas JSON adotam, em geral, uma representação própria do esquema no formato JSON, normalmente construções do tipo *chave:tipo_de_dado*. Apenas o trabalho de Kapsammer *et al*. [7] afirma utilizar a especificação JSON *Schema* [6] como modelo intermediário.

Com relação à representação final do esquema para documentos JSON também não há um consenso. Os trabalhos que usam a abordagem de transformação de modelos apresentam o esquema usando algum formato de metamodelo ECORE [5,7,12], normalmente tratado como metamodelo de domínio da aplicação. Outros trabalhos utilizam uma estrutura própria, como é o caso da *eSiBu-Tree* [14] e do JSON *DataGuide* [11], ou geram um esquema físico relacional [2]. Apenas em [9] percebe-se o uso efetivo da especificação JSON *Schema* [12]. Em [8], que trata de integração semântica, é adotada a OWL como modelo final de representação do esquema.

A falta de um esquema de dados explícito (*schemaless*) é uma característica atraente em bancos de dados NoSQL para desenvolvedores de aplicação. Em função disso, a evolução dos dados também é mais fácil porque não há necessidade da evolução

do esquema, fazendo com que diferentes versões de dados possam coexistir. A maioria dos trabalhos apresenta um único esquema ao final do processo, o qual busca representar todo o conjunto de dados existente na fonte de origem. No entanto, percebe-se que alguns trabalhos se preocupam em distinguir as diferentes variações de esquemas que os dados podem ter no banco de dados. Em [12] é proposto o versionamento de esquemas, sendo que as versões de esquemas permitem representar a evolução dos dados no tempo. Em [14] também é gerado um *record schema* para cada tipo de documento JSON, distinguindo as diferenças estruturais presentes na coleção. No entanto, os autores propõem o uso do conceito de *skeleton* para criar uma representação única do esquema para os documentos JSON, a qual contém os principais atributos da coleção, identificados através de uma medida de qualidade. Em [9] são apresentados esquemas únicos por coleção. Além disso, identificam-se atributos obrigatórios e opcionais e utiliza-se um grau de cobertura como medida para definir o grau de similaridade entre dois documentos JSON.

Com relação às etapas de desenvolvimento dos processos de extração de esquemas, percebe-se uma certa semelhança entre os trabalhos. Cada etapa representa desafios que precisam ser tratados e podem ser resumidos em: *a*) selecionar o conjunto de documentos JSON a ser considerado na geração dos esquemas (essa etapa é influenciada pela fonte de dados JSON); *b*) extrair o esquema de cada documento JSON e representá-lo em um modelo; *c*) inferir um esquema de domínio (único), por meio da integração dos esquemas, de cada documento JSON.

5. Considerações Finais

O volume, variedade e velocidade dos dados atuais impulsionou o surgimento de modelos de representação de dados complexos, como o formato JSON, e novas categorias de bancos de dados, como os bancos de dados NoSQL, os quais são fortemente caracterizados pela ausência de esquema. A falta de definição de esquema oferece uma maior flexibilidade, facilita a inclusão de dados não uniformes e a evolução dos dados. Neste cenário, percebe-se atualmente o interesse em extrair esquemas de dados JSON visando o desenvolvimento de ferramentas de gerência de dados ou a integração de bancos de dados distintos. O problema da extração de esquemas JSON é ainda uma questão em aberto na comunidade de banco de dados. Assim sendo, a contribuição principal deste trabalho é a apresentação e comparação de propostas que sintetizam soluções possíveis para este problema, visando servir como um guia para a pesquisa pelos interessados no assunto.

A partir do quadro comparativo apresentado pode-se inferir alguns desafios que precisam ser explorados na busca por soluções mais efetivas. Alguns deles podem estar relacionados ao domínio da aplicação, como, por exemplo:

- Seleção de registros: no contexto de *BigData*, em que coleções possuem um grande volume de dados, pode não ser viável usar todo esse volume em um processo de extração de esquemas. Dessa forma, é necessário definir estratégias para a seleção dos dados;
- Formato de representação de esquemas: o padrão JSON *Schema* ainda se encontra em desenvolvimento. Necessita-se verificar se ele já está em condições de atender as necessidades de representação ou se é melhor usar uma das alternativas propostas ou ainda definir novos padrões;

- Unificação de esquemas: a disponibilização de um único esquema ou vários esquemas que representem as mudanças estruturais nas coleções de dados JSON não tem uma resposta final. Técnicas e ferramentas para fazer *merge*/*matching* de esquemas precisam ser adaptadas para uso com JSON *Schema*.

Referências

[1] **DB Engines**. 2016. Disponível em: <http://db-engines.com/em/ranking_trend>. Acesso em: 07 dez. 2016.

[2] DISCALA, M.; ABADI, D. J. Automatic Generation of Normalized Relational Schemas from Nested Key-Value Data. In: INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA - SIGMOD, 2016. **Proceedings...** New York: ACM, 2016.

[3] GHAREHCHOPOGH, F. S.; KHALIFELU, Z. A. Analysis and evaluation of unstructured data: text mining versus natural language processing. In: INT. CONF. ON APPLICATION OF INFORMATION AND COMMUNICATION TECHNOLOGIES - AICT, 5., 2011. **Proceedings...** Azerbaijan, Baku, 2011.

[4] HAINAUT, J. L. *et al*. The Nature of Data Reverse Engineering. **Data Reverse Engineering Workshop**, EuroRef, Seventh Reengineering Forum, Reengineering Week 2000, Zurich, Switzerland, March 2000.

[5] IZQUIERDO, J. L. C.; CARBOT, J. Discovering implicit schemas in JSON data. **Lecture Notes in Computer Science**, v. 7977, Heidelberg: Springer-Verlag, 2013.

[6] JSON Schema Community. **JSON Schema**, 2016. Disponível em: <http://json-schema.org>.

[7] KAPSAMMER, E. *et al*. User profile integration made easy - Model-driven extraction and transformation of social network schemas. In: ANNUAL CONFERENCE ON WORLD WIDE WEB COMPANION – WWW, 21., 2012. **Proceedings...** 2012.

[8] KIRAN, V. K.; VIJAYAKUMAR, R. Ontology Based Data Integration of NoSQL Datastores. In: INDUSTRIAL AND INFORMATION SYSTEMS – ICIIS, 9., 2014. **Proceedings...** 2014

[9] KLETTKE, M.; STÖRL, U.; SCHERZINGER, S. Schema Extraction and Structural Outlier Detection for JSON-based NoSQL Data Stores. In: BTW. **Proceedings...** LNI.GI, 2015

[10] JSON.ORG. **Introducing JSON**. Disponível em: <http://json.org>. Acessado em: 17 dez. 2016.

[11] LIU, Z. *et al*. Closing the Functional and Performance Gap Between SQL and NoSQL. In: INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA – SIGMOD, 2016. **Proceedings…** San Francisco (Califórnia): ACM, 2016

[12]RUIZ, D. S.; MORALES, S. F.; MOLINA, J. G. Inferring versioned schemas from NoSQL databases and its applications. **Lecture Notes in Computer Science**, v. 9381, p. 467–480, 2015.

[13] STEINBERG, D. *et al*. **EMF - Eclipse Modeling Framework**. 2. ed. Boston: Pearson, 2009.

[14] WANG, L. Schema Management for Document Stores. **The VLDB Endowment**, v. 8, n. 9, p. 922–933, 2015.

Análise e Comparação de Algoritmos de Similaridade e Distância entre *strings* Adaptados ao Português Brasileiro

Diogo Luis Von Grafen Ruberto[1], **Rodrigo Luiz Antoniazzi**[2]

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

[2]Centro de Ciências Humanas e Sociais (CCHS)
Universidade de Cruz Alta (UNICRUZ) – Cruz Alta – RS – Brasil

diogo.rubert@gmail.com, rodrigoantoniazzi@yahoo.com.br

Abstract. *The use of databases in business is fundamental to decision talking, but the information extraction in DBMS could use techniques, to make a smart search. Searches that use relational operators are limited when there are typos or when the database is inconsistent. To correct this, some systems have functions that allow you to search based on the similarity of strings, for example, searches based on phonetic algorithms such as Soundex and Metaphone, but both methods are unusual in languages other than English and therefore need of an adaptation. The algorithm calculating the distance between strings, based on calculating the Levenshtein distance, another alternative is to find similarities between two strings. In this context, it is necessary to identify which algorithm is more efficient both in performance and in the accuracy of the data returned. Furthermore, it should be considered that the efficiency varies with the database, and if hybrid methods are the best alternative.*

Resumo. *A utilização bancos de dados nas empresas é fundamental para tomada de decisões, porém a recuperação de informações nos SGBD poderia utilizar técnicas para tornar as buscas mais inteligentes. As buscas que utilizam operadores relacionais são limitadas quando ocorrem erros de digitação ou quando a base de dados está inconsistente. Para suprir esta deficiência, alguns sistemas possuem funções que permitem fazer buscas baseadas na similaridade das strings, por exemplo, as buscas baseadas em algoritmos fonéticos como o Soundex e o Metaphone, porém ambos os métodos não são usuais em idiomas diferentes do inglês e precisam, portanto, de uma adaptação. O algoritmo do cálculo da distância entre strings, baseado no cálculo da distância de Levenshtein, é outra alternativa para encontrar similaridades entre duas cadeias de caracteres. Neste contexto, é necessário identificar qual algoritmo é o mais eficiente, tanto na performance quanto na precisão dos dados retornados. Além disso, deve ser analisado se a eficiência varia de acordo com a base de dados, e se os métodos híbridos são a melhor alternativa.*

1. Introdução

O armazenamento e relacionamento de informações em um Sistema Gerenciador de Banco de Dados (SGBD) são essenciais para uma organização, uma vez que grandes

bancos de dados podem conter informações importantes, as quais permitem as empresas tomar decisões e gerar conhecimento [Sudarshan et al. 2006].

A recuperação das informações é tão importante quanto o seu armazenamento e, por isso, deve ocorrer de maneira simples e precisa [Frantz 2009]. No momento em que um grande volume de dados é armazenado, a Lógica *Fuzzy* auxilia no reconhecimento de padrões para que estes dados se tornem informações úteis aos usuários [Chen et al. 2016]. Entretanto, a recuperação de informações pode ser trabalhosa por meio dos métodos tradicionais de comparação de *strings* dos SGBD atuais.

Apesar da eficiência do operador *like*, do operador de igualdade e demais operadores lógicos em consultas SQL, eles são limitados em bases de dados onde ocorreram erros de digitação ou em buscas fonéticas. Mesmo que o banco de dados esteja consistente e as informações cadastradas corretamente, a falha humana pode ocorrer no momento em que o usuário digita a informação que deseja buscar [Frantz 2009].

O erro humano não deve impedir que um sistema funcione corretamente e, por este motivo, as técnicas que serão abordadas têm inúmeras aplicabilidades. Por exemplo, em um hospital o paciente não poder deixar de ser atendido porque o usuário digitou "Amiuton" ao invés de "Hamilton". Em um sistema de vendas *on-line* o produto não pode deixar de ser enviado para a cidade correta porque o comprador digitou "Pajuçara" quando o correto seria "Pejuçara". Ao mesmo tempo, os métodos podem ser utilizados em corretores ortográficos [Piltcher et al. 2005] ou buscadores que sugerem a palavra correta ao usuário que cometer um erro.

Algumas linguagens de programação e Sistemas Gerenciadores de Banco de Dados (SGBD) disponibilizam funções nativas ou extensões que possibilitam a busca de informações com base na similaridade dos dados, como por exemplo, o cálculo da distância *Levenshtein* e os algoritmos fonéticos *Soundex* e *Metaphone*. Entretanto, conforme [PostgreSQL 2016] as funções fonéticas são pouco usuais em idiomas diferentes do inglês e, portanto, existe a necessidade de adaptar estes algoritmos para outros idiomas.

Além de adaptar as funções fonéticas para o português brasileiro, no presente estudo são aplicados métodos a fim de facilitar a recuperação de dados em grandes bases, mesmo que estes estejam inconsistentes, que tenham ocorrido erros ao informar o dado que se deseja encontrar ou até mesmo quando não se sabe exatamente a informação que deseja localizar. Após, é realizada uma comparação dos resultados obtidos com os métodos aplicados com objetivo de identificar qual deles foi mais eficiente na recuperação de informações. Também será comparado o desempenho para execução das funções, com a intenção de identificar qual o melhor método.

1.1. Organização do trabalho

Este trabalho está organizado em cinco sessões. A Sessão 2 trata da revisão bibliográfica sob a qual este artigo foi fundamentado. A Sessão 3, apresenta um estudo detalhado sobre algoritmos de similaridade e distância entre *strings*, em especial os algoritmos *Soundex*, *Metaphone* e *Levenshtein*.

A Sessão 4 trata da implementação das funções adaptada ao português brasileiro e do ambiente de testes que foi utilizado. Por fim, a Sessão 5 apresenta as considerações finais.

2. Revisão Bibliográfica

A seguir são apresentadas algumas pesquisas sobre similaridade e distância entre *strings*, fonética da língua portuguesa e problemas em consultas a bases de dados, que serviram como base para a elaboração desta pesquisa.

[Frantz 2009] descreveu as dificuldades encontradas na recuperação de informações, tais como: a redundância, inconsistência e ambiguidade dos dados, e destacou como a qualidade de uma aplicação pode ser comprometida com estes problemas. Analisou que as soluções disponíveis para solução destas questões são limitadas quando tratam de textos escritos em línguas diferentes do inglês. Fez um estudo dos fonemas da língua portuguesa e de algoritmos fonéticos, onde adaptou os métodos para o português brasileiro e criou uma função para recuperação de informações em bancos de dados. Por fim, elaborou um estudo comparativo entre o protótipo desenvolvido e uma ferramenta existente, e por meio de um questionário avaliou a eficácia do algoritmo. O estudo dos fonemas e a adaptação dos algoritmos fonéticos para o português brasileiro são de suma importância para este estudo, pois apresentam os resultados estatísticos e as dificuldades encontradas na adaptação destes métodos.

[Jardini 2012] propôs um ambiente *data cleaning* a fim de melhorar a qualidade dos dados armazenados em bancos de dados inconsistentes. No trabalho são discutidos os problemas de inconsistência e duplicidade de dados, além de expor diversas técnicas para identificação de similaridades, baseadas em caracteres, *token* e fonética. Entre as técnicas discutidas, registrou o uso dos algoritmos de *Levenshtein*, *Soundex* e *Metaphone*, e explicou o conceito de cada um. Para detecção de duplicidades e inconsistências, usou o algoritmo de *Levenshtein* e para permitir que a ferramenta identificasse duplicidade, independente do idioma, implementou uma detecção fonética multi-idioma. Também aplicou testes do ambiente e comprovou que a ferramenta cobriu aproximadamente 90% das inconsistências. As técnicas que serão analisadas na presente pesquisa foram conceituadas e testadas no trabalho correlato supracitado, por isso a sua importância.

[Borges 2008] apresentou um mecanismo de deduplicação de metadados e rastreio da proveniência. O sistema foi aplicado em bibliotecas digitais onde ocorrem problemas como variações de grafia e omissão de palavras. Para identificação da similaridade entre os títulos dos objetos digitais foram aplicadas técnicas baseadas no algoritmo de *Levenshtein*. Explicou que a técnica foi escolhida pois preserva a ordem em que as palavras aparecem em uma *string*, porém pode ter alguns problemas com caracteres acentuados. Foram definidas métricas de avaliação e aplicados experimentos para medir a precisão de cada algoritmo aplicado. A aplicação do algoritmo de *Levenshtein* e as métricas utilizadas devem auxiliar nas conclusões sobre a eficiência do método.

3. Algoritmos Fonéticos e de Similaridade

Nesta sessão serão abordados alguns algoritmos que fazem buscas inteligentes. Apesar das linguagens de programação e SGBD disponibilizarem funções fonéticas que retornam representações das palavras em forma de códigos, não existem funções específicas para língua portuguesa.

Os algoritmos fonéticos escolhidos para ser estudados e adaptados, foram o *Soundex* e o *Metaphone*, pois os mesmos são a base para os diversos outros existentes

[Croft et al. 2016]. O algoritmo de *Levenshtein* foi escolhido pois, mesmo que o usuário saiba escrever corretamente a palavra ou nome que deseja buscar, poderia ocorrer um simples erro de digitação que, mesmo que mude sua pronúncia, é muito próximo da palavra correta. O cálculo da distancia entre *strings* deve resolver este problema.

A intenção destes métodos é ir além da busca exata, aquela que utiliza operadores relacionais [Snae 2007]. Quando utiliza-se o operador "=" (igual), é necessário transcrever a *string* exatamente como ela está armazenada. Mesmo que se utilizasse a função *"upper"* para que a consulta não seja *case sensitive*, a necessidade de digitar a sequência de caracteres exatamente como ela está armazenada no banco de dados é uma premissa básica. Mesmo que fosse utilizado o operador *like* em um comando SQL para procurar um determinado padrão, chega-se a uma proximidade matemática, exata e previsível. Para encontrar o nome "Vilson" quando não se sabe se o correto é "Vilson", "Wilson", "Vilsom" ou "Wilsom" pode ser bastante trabalhoso.

3.1. *Soundex*

O código *Soundex* foi criado por Robert C. Russell e Margaret K. Odell em 1918. Inicialmente foi usado no censo americano como uma forma de indexar os nomes das pessoas. A ideia de Russell foi ordenar o nome das pessoas não por ordem alfabética, mas sim pela forma como era pronunciado. O código *Soundex* de uma palavra é representado pela letra inicial mais um código de três números que é obtido a partir de uma tabela [Binstock and Rex 1995].

A tabela de códigos *Soundex* criada por Russell foi baseada na classificação dos fonemas da língua inglesa [Reyes-Barragán et al. 2009]. Para adaptar o *Soundex* para o português brasileiro, a proposta do presente trabalho é mudar o valor da tabela de códigos baseado na classificação fonética língua portuguesa, mais precisamente nos pontos de articulação utilizados para pronunciar as consoantes, já que as vogais são ignoradas neste método. A tabela 1 apresenta o proposta para adaptação.

Tabela 1. Códigos *Soundex* para Português Brasileiro

Letra(s)	Valor	Pontos de Articulação
A, E, I, O, U, H, W, Y	0	-
P, B, M	1	Bilabiais
F, V	2	Labiodentais
T, D, N	3	Linguodentais
L, R	4	Línguo-Alveolares
S, Z	5	Línguo-Alveolares Convexas
J, DI, GI, TI, CH, LH, NH	6	Línguo-Palatais
K, C, G, Q	7	Velares
X	8	-

Como no português existem dígrafos, ou seja, duas consoantes que juntas formam um só fonema como observa-se nas palavras que possuem as consoantes "CH", "LH" e "NH", o algoritmo deverá ser adaptado para verificar estes encontros consonantais e tratar de forma adequada. As letras "W" e "Y" foram mantidas como letras a serem descartadas, pois são utilizadas apenas para representar palavras estrangeiras que ainda

não foram aportuguesas. Também por produzirem os sons de "U" e "I", respectivamente, as quais também são descartadas no *Soundex* assim como as demais vogais.

Entretanto, a letra "X", pode representar quatro fonemas na língua portuguesa [Chbane 1994]. É o caso das palavras exame, máximo, complexo e xícara. Percebe-se que em cada uma das palavras o "X" é pronunciado de forma diferente. Por esse motivo, ele será tratado como um código independente.

3.2. *Metaphone*

Assim como o *Soundex*, o algoritmo *Metaphone* também é considerado um algoritmo fonético. Ele foi escrito por Lawrence Philips em 1990 com o objetivo de suprir as deficiências do *Soundex*. Além desse método, o autor também desenvolveu os métodos *Double Metaphone* e *Metaphone* 3, que são melhoramentos do algoritmo original [Lisbach and Meyer 2013].

Como uma mesma letra pode representar mais de um som, a ideia do *Metaphone* é identificar a posição onde a letra está inserida, para assim definir a sua melhor representação. Outra definição deste método, é que aplicam-se regras para transformar um som em outro. Diferente do *Soundex*, não são consideradas apenas consoantes para definir uma representação fonética. As vogais também são importantes para identificar o som que um conjunto de caracteres pode representar [Binstock and Rex 1995]. Assim como o *Soundex*, o algoritmo *Metaphone* usa regras para fazer as transformações. Obviamente, estas regras foram baseadas na língua inglesa e não são usuais no português. Com estas regras, as palavras são transcritas para uma representação fonética, de maneira que palavras que soam de maneira semelhante serão representadas da mesma forma.

Baseado nas regras deste algoritmo fonético, [Jordão and Rosa 2012] escreveram um artigo sobre a importância da fonética na busca e correção de informações textuais. Neste artigo, apresentaram uma proposta de adaptação para o português brasileiro, denominado *Metaphone-pt_BR*. Os autores explicam que obtiveram resultados satisfatórios com as novas regras, entretanto, mesmo com a adaptação, o algoritmo é eficiente com palavras encontradas no dicionário, mas em nomes e sobrenome em que a pronuncia depende do local de origem, deveriam ser usadas regras específicas.

3.3. *Levenshtein*

O conceito da distância de *Levenshtein* foi escrito em 1965 pelo matemático Vladimir I. Levenshtein e baseado na distância de *Hamming*, porém, a diferença é que pode ser usado para comparar palavras com tamanhos diferentes. O princípio de *Levenshtein* é definir a distância entre duas palavras com base no número de operações necessárias para torná-las iguais. Cada operação tem um determinado custo que é acumulado de acordo com as edições realizadas: inserção, exclusão ou substituição [Lisbach and Meyer 2013].

[Wagner and Fischer 1974] observaram que para calcular a distância de *Levenshtein* seria necessário no mínimo $m * n^2$ comparações e, então, publicaram um algoritmo capaz de reduzir esta complexidade para $m * n$, conhecido como *edit-distance*. Apesar da sua utilidade, este algoritmo pode ser lento para comparar *strings* muito longas, pois a matriz que precisa ser criada é diretamente proporcional ao tamanho de cada *string*.

4. Ambiente Experimental

Após conceituar os algoritmos de *Levenshtein*, *Soundex* e *Metaphone*, e definir as adaptações necessárias, foram implementadas em *PL/pgSQL* as funções abaixo. Esta linguagem foi escolhida por tratar-se de uma extensão do padrão SQL que permite a implementação de funções robustas no *PostgreSQL*. Este SGBD, por sua vez, foi escolhido por ser *opensource*, robusto e com uma boa documentação.

- *br_levenshtein*: função para calcular a distância de *Levenshtein* baseada no algoritmo de Wagner e Fisher (1974);
- *br_soundex*: função para calcular o código *Soundex* adaptado para o português brasileiro, conforme proposto neste trabalho;
- *br_metaphone*: função para calcular o código *Metaphone* adaptado para o português brasileiro, conforme proposto por [Jordão and Rosa 2012].

4.1. Métricas Utilizadas

Conforme [Sudarshan et al. 2006], para medir a eficácia de funções que recuperam informações, podem ser aplicadas medidas de precisão e de revocação. Assim, os algoritmos foram comparados entre si, e efetuadas estas medidas a partir da geração de dados estatísticos e gráficos. Também foi analisado se a hipótese de que soluções híbridas tem um resultado melhor, é verdadeira.

Os testes realizados a seguir utilizaram as seguintes métricas para avaliação dos resultados.

- Precisão: quando executada uma consulta, deve medir a taxa de acerto da função, isto é, quantos registros foram retornados corretamente em relação a todos os registros retornados;
- Revocação: quando executada uma consulta, deve medir a taxa de registros relevantes retornados, ou seja, quantos registros foram retornados corretamente em relação ao total de registros que a função deveria de fato retornar;
- Medida F Balanceada: é a média harmônica ponderada da precisão e da revocação. Será utilizada para medir a relação entre as duas métricas utilizadas;

4.2. Testes e Resultados

Os testes das funções foram executados em bases que simulam várias situações de buscas. Um banco de dados com 5.500 registros com nomes de cidades. Outro, com 310.000 registros com palavras da língua portuguesa. E, por fim, uma base com 10.000 registros com nomes de pessoas e empresas.

A análise e comparação foi efetuada a partir da coleta de algumas amostras de cada base de dados, as quais determinaram quais dados deveriam ser retornados. A partir disso, foi calculado o percentual médio de todas as consultas. Na base com 5.500 registros foram realizadas 25 consultas para cada um dos tópicos. Na base com 10.000 registros, 50 consultas. Já na base com 310.000 registros, foram realizadas 100 consultas.

4.2.1. Base de dados com nomes de cidades

O primeiro teste utilizou uma base de dados com o nome de cidades, cujo objetivo foi analisar a eficiência das funções em situações onde os dados não sofreram erros de digitação,

e que os operadores lógicos também encontrariam. Mas também procurou analisar a situação em que o dado inicial foi digitado incorretamente, porém o erro não alterou a forma como a palavra é pronunciada. Para exemplificar, poderia-se tentar buscar a cidade de "Chiapeta" onde o correto é "Chiapetta". Da mesma forma, fez-se a verificação que utiliza dados inconsistentes, isto é, caracteres faltando e caracteres que alteram, em partes, forma como uma palavra é pronunciada.

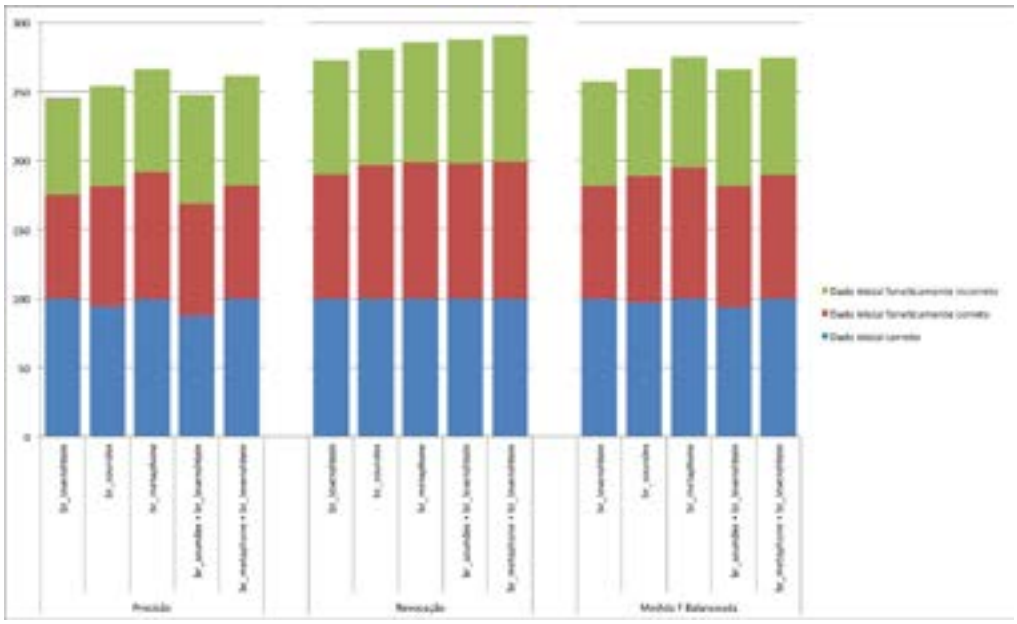

Figura 1. Gráfico da base de dados com nomes de cidades

Nesta base, foi observado que nenhuma função deixou de mostrar uma informação relevante quando o dado inicial estava correto, pois atingiram 100% de revocação. Por outro lado, os métodos *br_soudex*, *br_metaphone* e o híbrido *br_soundex* + *br_levenshtein* retornaram alguns dados a mais, o que reduziu a precisão. De qualquer forma, o resultado da média harmônica ponderada foi satisfatório, e ficou acima dos 90%.

Pode-se observar na Figura 1 que as funções fonéticas superam o método da distância entre strings quando dado inicial sofre alterações que não alteram a sua pronúncia. Observa-se ainda que a função *br_metaphone* obteve a melhor média ao relacionar precisão e revocação.

4.2.2. Base de dados com nomes de pessoas e empresas

Este teste utilizou uma base de dados com nomes de pessoas e empresas, onde tem-se praticamente o dobro de registros da base utilizada no teste anterior. O objetivo foi verificar a eficácia e performance em uma base com maior volume de registros e que nomes estrangeiros podem não representar necessariamente um fonema da língua portuguesa. A intenção do teste é simular situações onde o usuário não sabe escrever corretamente o sobrenome de uma pessoa, por exemplo. Da mesma forma que o teste anterior, este experimento fez consultas para simular a inconsistência em bases que armazenam nomes. São situações em que um nome está armazenado incorretamente ou quando houve uma falha na digitação. Os resultados são demonstrados na Figura 2.

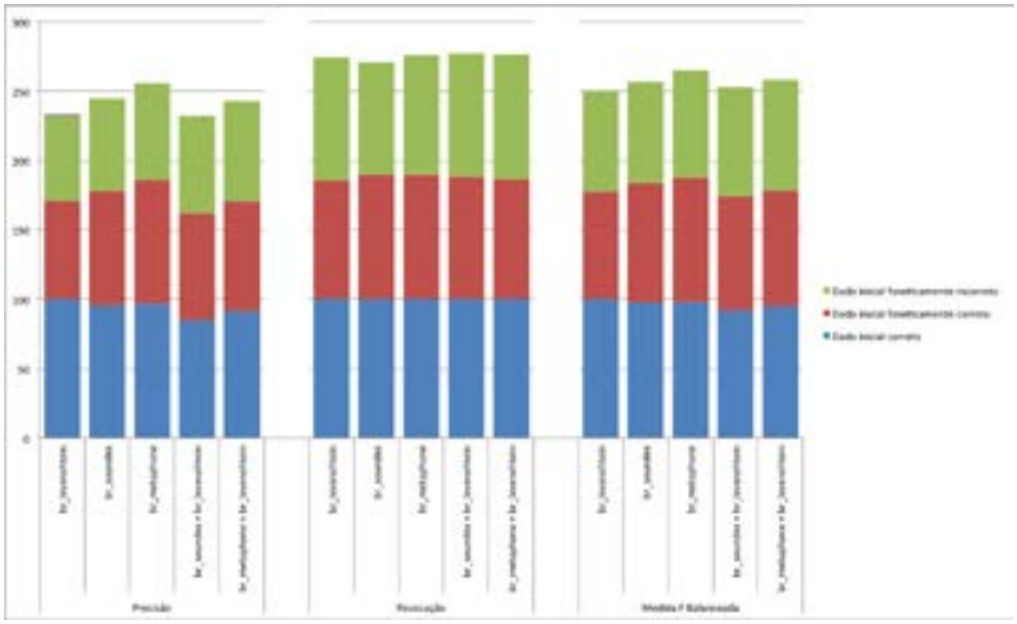

Figura 2. Gráfico da base de dados com nomes de pessoas e empresas

No caso dos nomes de empresas e pessoas, as funções fonéticas ainda são superiores, porém verificou-se uma pequena queda nos percentuais de precisão e revocação, consequentemente, na medida F balanceada. Isto explica-se pois alguns sobrenomes estrangeiros não são pronunciados exatamente da forma que foram escritos. Cabe lembrar que os métodos foram adaptados apenas para o português brasileiro.

4.2.3. Base de dados com palavras do dicionário

A terceira e última análise, utiliza uma base com 310.000 registros que, na verdade, trata-se de um dicionário da língua portuguesa. O objetivo do teste foi determinar efetividade das funções com grandes volumes de dados em um ambiente que contém a maioria das palavras utilizadas habitualmente na língua portuguesa. Este ambiente de testes representa a situação de um corretor ortográfico, que precisa analisar o dado digitado e retornar sugestões ao usuário.

Ao avaliar os métodos neste experimento, verificou-se que a média para a função *br_metaphone* mantem-se constante e é superior as demais, e que as soluções híbridas tiveram bons resultados como pode ser verificado na Figura 3. Apesar da precisão ser pequena, pois retornou uma série de informações desnecessárias, a revocação teve um bom percentual, afinal os dados que precisavam ser mostrados foram apresentados corretamente.

No quesito revocação, pode-se afirmar que todas as funções são totalmente eficientes, pois nenhum registro deixou de ser listado. Em relação à precisão, apenas a função *br_levenshtein* foi totalmente precisa. As demais retornaram alguns registros a mais, porém chegaram muito próximo dos 100%.

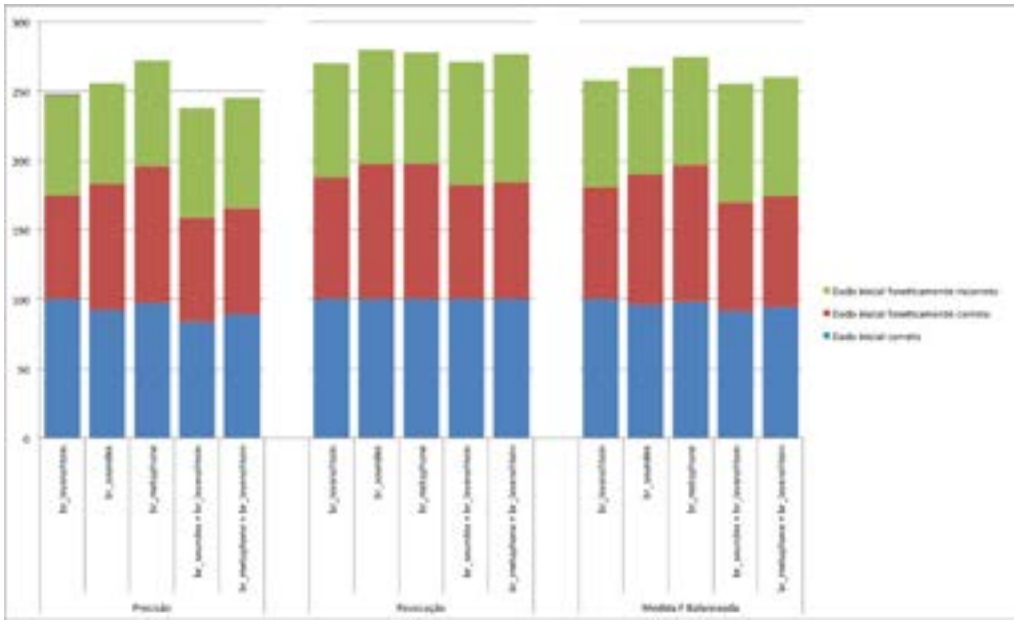

Figura 3. Gráfico da base de dados com palavras do dicionário

5. Considerações Finais

Neste estudo procurou-se comprovar que as adaptações dos métodos fonéticos são possíveis para a língua portuguesa e responder qual o melhor método para recuperação de informações. A compreensão da fonética, fonologia e fonemas da língua portuguesa foi fundamental para a adaptação dos métodos *Soundex* e *Metaphone*, pois ainda existem poucos estudos para o português brasileiro.

Quando o dado inicial está correto, todos os métodos atingiram 100% no quesito revocação, isto é, nenhum registro que deveria ser listado deixou de ser apresentado. Entretanto, o objetivo deste trabalho é ir além das buscas exatas utilizando operadores lógicos. Por este motivo, os testes com o dado inicial incorreto devem ter maior relevância na análise dos resultados.

No quesito precisão, as funções fonéticas demonstraram ser bastante eficientes quando ocorrem erros de digitação. Destaca-se que a função *br_metaphone* se sobressai em relação as demais. Os dados retornados por esta função foram bastante precisos e chegaram à 99,1% de precisão com o dado inicial foneticamente correto. Outro fator observado, é que o uso da função *br_metaphone* em uma solução híbrida traz resultados superiores a mesma solução híbrida utilizando *br_soundex*. Já a função *br_levenshtein*, foi precisa apenas quando o dado inicial estava correto. Nos demais casos, foi pouco precisa com médias entre 61,4% e 75,0%.

As funções de similaridade mostraram ser uma alternativa interessante para suprir as limitações dos operadores lógicos. O uso destas técnicas podem ter aplicabilidades em inúmeros tipos de sistema, pois possibilitam uma busca alternativa ao usuário e permite a sugestão de informações, situação esta que vimos em buscadores modernos e corretores ortográficos. Entretanto, os métodos estudados são eficientes apenas com palavras do dicionário, e perdem bastante eficiência na busca de nomes, sobrenomes e palavras estrangeiras.

Como sugestão de trabalhos futuros, pode-se comparar a utilização de outras técnicas de detecção de similaridade entre *strings*, sejam elas baseadas em caracteres, *token* ou fonética. A adaptação destas funções para ambientes multi-idioma também poderia ser pesquisada. Estes métodos também podem ser usados para criação ou melhoria de sistemas de reconhecimento da fala.

Referências

Binstock, A. and Rex, J. (1995). *Practical algorithms for programmers*. Addison-Wesley Longman Publishing Co., Inc.

Borges, E. N. (2008). Md-prom: um mecanismo de deduplicação de metadados e rastreio da proveniência. Master's thesis, Universidade Federal do Rio Grande do Sul.

Chbane, D. T. (1994). *Desenvolvimento de sistema para conversão de textos em fonemas no idioma português*. PhD thesis, Universidade de São Paulo.

Chen, S.-M., Cheng, S.-H., and Lan, T.-C. (2016). A novel similarity measure between intuitionistic fuzzy sets based on the centroid points of transformed fuzzy numbers with applications to pattern recognition. *Information Sciences*, 343:15–40.

Croft, D., Brown, S., and Coupland, S. (2016). An effective named entity similarity metric for comparing data from multiple sources with varying syntax. *Digital Scholarship in the Humanities*, page fqw035.

Frantz, R. R. R. (2009). Recuperação de informações por similaridade de fonemas adaptada à lingua portuguesa. *Centro Universitário Ritter dos Reis*.

Jardini, T. (2012). Ambiente data cleaning: suporte extensível, semântico e automático para análise e transformação de dados. Master's thesis, Universidade Estadual Paulista (UNESP).

Jordão, C. C. and Rosa, J. L. G. (2012). Metaphone-pt_br: the phonetic importance on search and correction of textual information. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 297–305. Springer.

Lisbach, B. and Meyer, V. (2013). *Linguistic identity matching*. Springer.

Piltcher, G., Borges, T., Loh, S., Lichtnow, D., and Simoes, G. (2005). Correçao de palavras em chats: Avaliaçao de bases para dicionários de referência. *XXV Congresso da Sociedade Brasileira de Computação, São Leopoldo: UNISINOS*.

PostgreSQL (2016). Postgresql 9.6.1 documentation. *Diponível em https://www.postgresql.org/docs/9.6/static/index.html, Acesso em: 09 nov. 2016*.

Reyes-Barragán, M. A., Pineda, L. V., and Montes-y Gómez, M. (2009). Inaoe at qast 2009: Evaluating the usefulness of a phonetic codification of transcriptions. In *CLEF (Working Notes)*.

Snae, C. (2007). A comparison and analysis of name matching algorithms. *International Journal of Applied Science. Engineering and Technology*, 4(1):252–257.

Sudarshan, S., Silberschatz, A., and Korth, F. H. (2006). Sistemas de banco de dados. 5ª. edição.

Wagner, R. A. and Fischer, M. J. (1974). The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173.

Redblock: Uma ferramenta para a deduplicação de grandes bases de dados em tempo real

Luan Félix Pimentel[1], **Igor Lemos Vicente**[1], **Guilherme Dal Bianco**[1]

[1]Universidade Federal da Fronteira Sul (UFFS)
Caixa Postal 101 – 89802-112
Departamento de Ciência da Computação
Chapecó, SC - Brasil.

luanfelixpimentel@gmail.com, guilherme.dalbianco@uffs.edu.br

Abstract. *Online data blocking aims to identify records that represent the same purpose on a streaming data environment. Real-time data blocking must be able to process a range of informations with a high effectiveness and no delays. The purpose of this paper is to introduce a developed tool named Redblock for real-time data deduplication using a distributed platform for online processement combined with a data blocking technique that make use of an Inverted Index. The tool managed to provide good preliminary results in terms of effectiveness in a synthetic database.*

Resumo. *A blocagem online de dados tem como propósito identificar registros que representam um mesmo objetivo em ambientes com fluxo contínuo de dados. A blocagem online deve ser capaz de processar volumes variados de informações, sem atrasos e com uma alta eficácia. Este trabalho, propõe uma ferramenta intitulada Redblock para a deduplicação de dados em tempo real. A ferramenta utiliza uma plataforma distribuída de processamento online em conjunto com um método de blocagem utilizando índice invertido. Na experimentação, Redblock demonstrou bons resultados preliminares em relação a sua eficácia em uma base de dados sintética.*

1. Introdução

A integração de dados tem como objetivo facilitar o acesso a informações a partir da consolidação de diferentes fontes de dados em um único repositório. Serviços como bibliotecas virtuais, *media streaming* e redes sociais dependem de um processo de integração com uma alta qualidade. Para isto, uma tarefa fundamental é a identificação de entidades (registros, documentos, textos, etc.) que já estão armazenadas na base de dados, portanto não devem ser novamente inseridas. Tal etapa, conhecida como deduplicação, tem por objetivo melhorar a qualidade das bases de dados, identificando entidades duplicadas.

A deduplicação envolve três etapas principais: blocagem, comparação e a classificação [Christen 2012]. A blocagem corresponde ao processo de geração de pares candidatos. Ou seja, todos os registros devem ser analisados em busca de potenciais duplicatas. Somente registros pertencentes a um mesmo bloco são utilizados para a criação dos pares candidatos com custo quadrático de processamento. Na etapa de comparação, a partir do conjunto de pares candidatos é computada a similaridade de cada par utilizando funções de similaridade (ex: Jaccard, Jaro, Ngram) [Mitra et al. 2005]. Por fim, na etapa

de classificação são utilizados algoritmos de classificação (árvores de decisão, máquinas de vetores de suporte, *Naive Bayes*, entre outros [Manning et al. 2008] ou heurísticas configuradas a partir de limiares manualmente definidos, para identificar quais pares representam uma duplicata.

Tradicionalmente, a deduplicação é tratada em bancos de dados estáticos (ou em lote). Isto quer dizer que a base de dados é processada, na sua totalidade, em busca de registros que representam dados duplicados. No entanto, o crescimento de serviços e aplicações com fluxo contínuo de dados em tempo real impulsiona a demanda por soluções capazes de suportar tal fluxo de dados. A deduplicação online, diferente da versão estática, deve ser capaz de lidar com picos de processamento sem que sejam evidenciados gargalos e ao mesmo tempo deve ser capaz de se adaptar a possíveis alterações nos padrões dos dados.

Este artigo propõe uma ferramenta, intitulada *Redblock*, para a deduplicação de dados online utilizando a plataforma *Apache Storm*. Essa plataforma é um sistema de computação distribuído e tolerante a falhas, que permite processar dados em tempo real. O principal desafio da *Redblock* é garantir que o maior número possível de pares duplicados sejam identificados. Apesar da *Redblock* implementar as principais etapas de deduplicação (blocagem e classificação), a principal contribuição é no processo de blocagem, no qual utilizamos um índice invertido que é armazenado em um banco de dados chave-valor. O objetivo desse banco de dados é otimizar o processo de busca das informações necessárias para a deduplicação dos dados. Para o processo de classificação, a *Redblock* utiliza um algoritmo supervisionado baseado em árvore de decisão. Tal algoritmo, por ser supervisionado, depende de um treinamento que deve ser minimizado o máximo possível, devido ao custo de rotulação. Dessa forma, a ferramenta utiliza uma amostragem aleatória para compor o treinamento do método.

O presente trabalho está organizado da seguinte forma: Na segunda seção é apresentados o embasamento para o entendimento da *Redblock*. Na seção 3 é descrito o funcionamento da ferramenta. Na quarta seção são exemplificados alguns trabalhos relacionados envolvendo o processo blocagem online. Na seção 5 apresentamos os experimentos e resultados obtidos pela terceira seção. Por fim, são apresentadas as considerações finais e como serão os próximos passos de evolução e aprimoramento da ferramenta.

2. Referencial Teórico

Com o objetivo de efetivar o entendimento do trabalho proposto, nesta seção serão apresentados alguns métodos presentes na literatura envolvendo a blocagem de dados. Em seguida, serão descritos os principais conceitos envolvendo o *framework* de processamento online *Apache Storm*, utilizado para o desenvolvimento da *Redblock*.

2.1. Técnicas de Blocagem

Um método mais simplista de produzir grupos ou blocos de dados é definir um dos atributos como o critério ou chave de blocagem. Somente registros que apresentarem uma mesma chave de blocagem serão inseridos em um mesmo bloco, reduzindo substancialmente o número de comparações. No entanto, em situações em que o atributo chave apresentar variações ou erros, pares duplicados poderão não ser agrupados corretamente reduzindo assim a qualidade do método [Elmagarmid et al. 2007].

Já a técnica de *Blocagem por Vizinho Ordenado* (BVO) busca ordenar a base de dados de acordo com os valores-chave dos blocos e sequencialmente move uma janela de tamanho fixo que irá criar os grupos de dados sobre os valores. Inicialmente, a base de dados é ordenada a partir do valor-chave selecionado (por exemplo, o atributo nome em uma tabela de cadastro) e em seguida, é organizado em ordem alfabética. A partir disso, os pares candidatos são gerados utilizando uma janela de tamanho fixo. Na Tabela 1 é apresentado um exemplo de base de dados contendo 4 registros, que ao aplicar uma janela deslizante com tamanho 3 produz um primeiro bloco contendo os primeiros 3 registros da base de dados (R1, R2 e R3) e outro bloco contendo os registros (R2, R3 e R4), conforme ilustrado na Tabela 2.

Identificador	Nome
R1	João Silva
R2	João
R3	Jo
R4	Teresinha Souza

Tabela 1. Exemplo de dados para aplicação da Indexação por vizinho ordenado.

Intervalo	Pares Candidatos
R1 - R3	(R1, R2), (R2,R3), (R1,R3)
R2 - R4	(R2,R3), (R3,R4), (R2,R4)

Tabela 2. Resultado da Indexação por vizinho ordenado, utilizando os registros da Tabela 1.

Um problema identificado por [Christen 2012] no método BVO é que a ordenação dos valores chave dos blocos são sensíveis perante erros e variações nas primeiras posições dos valores. Por exemplo, se a base de dados fosse ainda maior, ”Christina”e ”Kristina”estariam bem distantes na lista organizada em ordem alfabética, mesmo sendo nomes bem semelhantes que talvez se refiram a mesma pessoa. Para contornar esse problema o método *Q-Gram* utiliza como chave de blocagem *substrings* de um atributo com tamanho *Q*. Os valores-chave do bloco geram *strings* utilizando os *grams* que se tornarão o valor-chave em um índice e seus componentes comparados posteriormente. Apesar do método *Q-Gram* obter bons resultados, o alto custo de se indexar substrings [Baxter et al. 2003] limita a aplicação do método a um atributo em específico.

Por fim, o método de Índice Invertido, utiliza palavras como forma de indexação de uma coleção de registros. As nomenclaturas utilizadas por esse método são separadas por Vocabulário (diferentes palavras do documento) e Ocorrência (frequência que determinada palavra aparece no documento). Abaixo, exemplificamos na Tabela 3 como estariam organizados os dados recebidos, com registro separado por vírgulas. Já na Tabela 4, apresentamos uma simples visualização do resultado da aplicação do método do Índice Invertido, usando como referência a primeira linha da tabela 3. Como pode ser notado, o método foi capaz de agrupar o registro 3 junto do restante, utilizando como critério o nome da rua.

Docs	Texto
1	João Silva, (49) 987453214, Avenida Gusmão Freire
2	João, (49) 987453214, Avenida Gusmão Freire
3	Jo Silvo, (49) 987453218, Avenida Freire

Tabela 3. Exemplo de dados com base para aplicação do Índice Invertido.

Número	Termo	Docs
1	João	1, 2
2	Silva	2
3	(49)987453214	1,2
4	Avenida	1,2, 3
5	Gusmão	1,2
6	Freire	1,2,3

Tabela 4. Organização dos dados após aplicação do Índice Invertido utilizando as informações do Doc 1, Tabela 3.

No índice invertido, pode-se perceber que sua metodologia de aplicação é similar ao método tradicional da blocagem. Neste método, não escolhemos um determinado campo e com base nele produzimos índices. Pelo contrário, aqui utilizamos todos os campos e geramos índices para cada registro que foi processado [Christen 2012]. Devido a esta flexibilidade tal método de blocagem foi explorado pela ferramenta *Redblock*, conforme será descrito na Seção 3.

2.2. Apache Storm

A plataforma *Apache Storm* possibilita desenvolver aplicações que demandem processamento massivo de dados em tempo real. É utilizada uma metodologia própria que permite que uma coleção de tuplas (lista de valores) sejam distribuídas e processadas por *spouts e bolts*. Os *spouts* são similares a processos com a função de realizar a leitura de uma fonte de dados. Já os *bolts*, processam as tuplas recebidas para serem distribuídas a outros *bolts* ou armazenam as informações em fontes externas. Os *bolts e spouts* são integrados a partir de uma topologia que possibilita conectar um *bolt* e *spouts* a fim de se formar uma arquitetura. Como o objetivo é o processamento contínuo de dados, a topologia será executada até que o usuário interrompa o processo.

A Figura 1 ilustra um exemplo de topologia com 2 *spouts* e 5 *bolts*. No exemplo, os *spouts* recebem os dados e os distribui para os outros 3 *bolts* seguintes. Esses, por sua vez, irão realizar um processamento sobre os dados e encaminharão para outros 2 *bolts* finais. Estes serão responsáveis por transformar e processar os dados recebidos do *bolt* anterior e salvar as informações necessárias[1].

[1]Informações fornecidas pela Apache Storm. Disponível em: $< http : //storm.apache.org/ >$. Acesso em 6 de fevereiro de 2017.

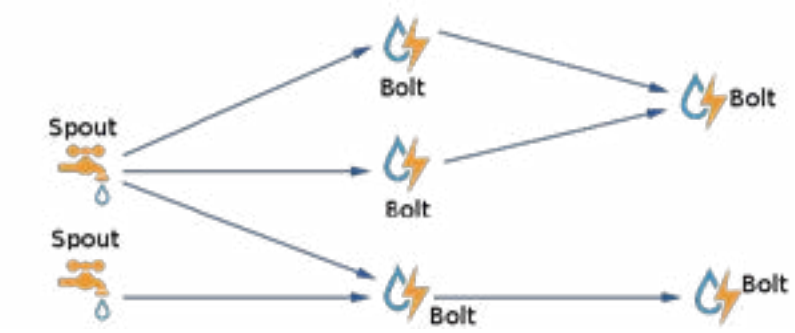

Figura 1. Exemplo básico do funcionamento de *Bolts* e *Spouts* em *Apache Storm*

3. Proposta - *Redblock*

Nesta seção, será descrita a ferramenta proposta para a deduplicação de dados em tempo real, denominada *Redblock*, com o objetivo de desenvolver um eficiente método para deduplicação online. A *Redblock* combina o método do índice invertido para o agrupamento de dados com a plataforma de processamento online *Apache Storm* para obter elasticidade no processamento massivo de dados.

Na Figura 2, ilustramos a topologia proposta. A *Redblock* é composta por 7 etapas principais com o objetivo de fragmentar o processamento, possibilitar que sejam identificados gargalos de processamento e tratados através do aumento do nível de paralelismo. Por exemplo, se o *bolt* que trata do processo de blocagem sofrer sobrecarga, é possível aumentar o número de *bolts* para esta tarefa para evitar atrasos de processamento. As linhas pontilhadas representam processos que ocorrem em paralelo no funcionamento da ferramenta, sendo respectivamente os *Bolts* que salvam dados em memória e o *Bolt* de treinamento que é iniciado previamente ao *Decision Tree Bolt* e ao *Counter Bolt* uma única vez. A seguir cada uma das etapas é descrita.

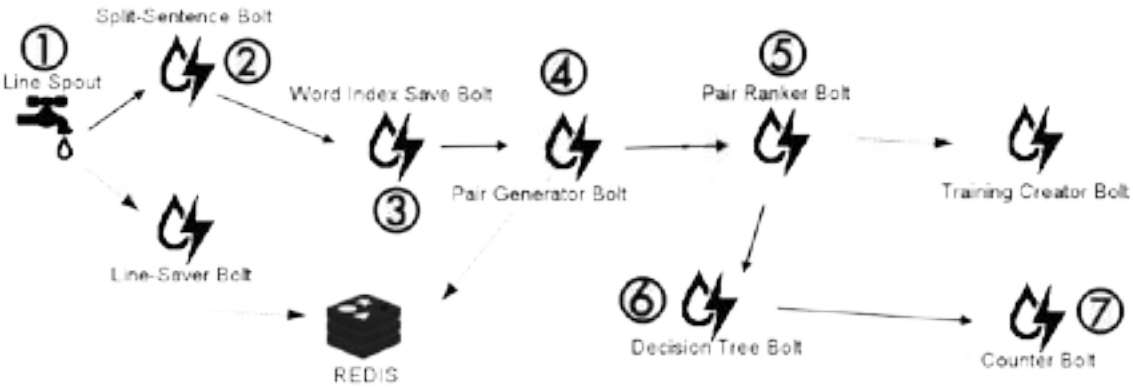

Figura 2. Topologia da ferramenta baseada na metodologia do *ApacheStorm*. Os números junto as etapas, ilustram o fluxo principal de execução da Redblock.

3.1. Spout - *Line Spout*

O *Line Spout* (1) representa o passo inicial da topologia. Ele desempenha a leitura da base de dados. Note que para desenvolver um ambiente de teste, o fluxo contínuo de dados é simulado através do carregamento de uma base de dados.

Após a leitura do arquivo, o *Line Spout* processa linha a linha da base de dados e envia a tupla gerada para o tratamento posterior do *bolt* destinatário. Por exemplo, a base de dados da Tabela 5 é carregada e enviada respectivamente para os *Bolts Line Saver* e *Split Sentence*. Note que uma cópia de cada registro é enviada para o *Bolt Line Saver* e outra cópia para o *Bolt Split Sentence*.

Registros do arquivo de entrada
01, joão, 6562348, estudante
02, joão, 98542138, professor
03, roberto, 52132157, estudante

Tabela 5. Exemplo de dados de entrada utilizados pela ferramenta.

3.2. Bolt - *Line Saver*

Como o processamento dos dados é em tempo real, um processo (ou um bolt) não congela enquanto outro está sendo executado. Portanto, o *Line Saver* realiza sua função em paralelo ao *Split Sentence Bolt*, recebendo a tupla e fazendo a sua tarefa.

O *Line Saver* tem como objetivo salvar a linha no banco de dados não relacional *REDIS*, armazenando os dados em memória para uma otimização da recuperação posterior. Ou seja, a tupla enviada pelo *spout* é salva no formato [ID][Linha].

3.3. Bolt - *Split Sentence*

Prosseguindo com o fluxo contínuo, o *Split Sentence Bolt* (2) recebe como entrada uma linha original (conforme a Tabela 5) e promove o processo de fragmentação. Ou seja, cada um dos atributos é emitido para a etapa seguinte para posterior indexação. Por exemplo, o primeiro registro da Tabela 5 é fragmentado em 3 partes: ”joão”, ”6562348”e ”estudante”e enviado para a próxima etapa juntamente com o ID de referência. Todos os campos serão emitidos para o *bolt* seguinte no formato [ID][campo].

3.4. Bolt - *WordIndex Save*

No *bolt- WordIndex Save* (3) é iniciado o processamento dos dados que são recebidos utilizando o método do índice invertido. Especificamente, a tupla enviada pelo *bolt Split Sentence* é recebida e processada de maneira que é criado um conjunto para cada palavra presente na base de dados. Isso evita a presença de termos duplicados. Caso a palavra já esteja no set, o ID de referência que veio na tupla é adicionado como uma lista de IDs que contém a mesma palavra. Abaixo, é ilustrado um exemplo simplificado do resultado do índice invertido das palavras ”João”e ”Silva”. Note que o termo ”joão”aparece nos registros com ID 01 e 02, já o termo ”6562348”está presente no registro com ID 01.

É possível notar a partir deste *bolt* pode-se identificar, através da simples contagem de ocorrências dos termos no índice invertido, a presença de palavras muito frequentes (conhecido como *stop word* [Manning et al. 2008]). RedBlock é definido um limiar para descartar tais termos. Por exemplo, se um termo estiver presente em mais de 50 registros este é ignorado pelo índice invertido.

Campo	Lista de IDs que contém o campo
João	01, 02
6562348	01
estudante	01,03

Tabela 6. Exemplo do *Bolt WordIndex Save* utilizando como entrada os registros da Tabela 5.

3.5. Bolt - *Pair Generator*

A função do *Pair Generator* (4) é de construir pares baseados no conjunto de palavras que foi salvo no banco de dados pelo *Bolt WordIndex Save*. A idéia é que todos os registros que compartilham uma palavra em comum devem ser comparados para verificar se representam uma duplicata. As principais funções desse *bolt* são recuperar o registro por completo (em sua totalidade) acessando o banco de dados e enviar os pares para a etapa seguinte.

Por exemplo, se o termo ”João”consta nos registros 01, 02 e 03, cada registro servirá de fonte para que o *Bolt Pair Generator* recupere o registro e envie para o próximo *bolt* os pares (1,2) e (1,3), conforme Tabela 6.

3.6. Bolt - *Pair Ranker*

O *bolt Pair Ranker* (5) tem como objetivo computar a similaridade de cada linha recebida pelo *bolt Pair Generator*, mensurando o grau de semelhança de cada atributo a partir de uma função de similaridade. As funções de similaridade utilizadas são as de *Jaccard* e *Leveshtein* [Manning et al. 2008] . Na função *Jaccard* é avaliado o nível de distância que duas *strings* se encontram. A distância *Levenshtein* entre duas *strings* é definida como o número mínimo de formatações necessárias para se transformar tal *string* na outra que está sendo comparada, sendo tais formatações representadas por inserção, substituição e eliminação de um ou mais caracteres.

3.7. Bolt - *Training Creator*

O objetivo do *bolt Training Creator* é construir um modelo de treinamento a partir do algoritmos de árvore de decisão. O modelo será utilizado posteriormente para identificar os pares duplicados e não duplicados. Note que o treinamento é configurado a partir de uma base de treinamento contendo um conjunto, de tamanho pré-definido, de pares rotulados pelo usuário. Dessa forma, identificar tais pares e rotulá-los é uma tarefa custosa que deve ser minimizada o máximo possível.

3.8. Bolt - *Decision Tree*

O *bolt Decision Tree* (6) utiliza o modelo de classificação, previamente criado, para identificar os pares como duplicatas ou não duplicatas. O *bolt* recebe como entrada os valores de similaridade de cada par e emite como saída a instância classificada. O algoritmo utilizado foi o J48.

3.9. Bolt - *Counter Bolt*

O último *bolt* presente na topologia é o *Counter Bolt* (7). Enquanto a topologia está em funcionamento, este bolt irá gerar ao usuário quantos pares Verdadeiro-Positivos,

Verdadeiro-Negativos, Falso-Positivos e quantos Falso-Negativos foram computados. Este bolt só terá funcionalidade na presença do gabarito da base de dados.

4. Trabalhos Relacionados

Um simples processo de deduplicação de dados pode ser realizado comparando todos os registros contra todos os outros presentes na base de dados. No entanto, tal método resulta em um custo quadrático de processamento, o que seria inviável no caso de uma base de dados média ou grande. Nesse contexto, a blocagem surge como uma alternativa para se reduzir o espaço de busca, somente processando os registros que possuem algum indício de representar uma duplicata.

No cenário de bases de dados incrementais, na qual os registros são inseridos ao longo do tempo, é importante que o método de deduplicação seja capaz de se autoajustar a novos padrões e reduzir ao máximo o tempo de processamento de uma nova entrada para atender a demanda online. Dessa forma, é importante que o processo de blocagem, que representa a maior fatia de processamento [Dal Bianco et al. 2015], seja suficientemente eficiente para não resultar em atrasos de processamento.

Um dos primeiros trabalhos envolvendo a deduplicação online foi proposto por [Bhattacharya and Getoor 2007], na qual é desenvolvido um método para que consultas envolvendo dados deduplicados sejam respondidas em tempo real. No método, dados são pré-processados para possibilitar a resposta em tempo real. Já no contexto da blocagem online, em [Bizer et al. 2009] é proposta uma técnica na qual as similaridades entre os atributos são pré-calculadas quando um índice invertido é criado. Dessa forma, o método depende de que boa parte dos dados estejam presentes estaticamente para que o índice seja populado apropriadamente, diferente do contexto online. Já [Ramadan et al. 2013] aprimora a estrutura de índice proposta por [Whang et al. 2009] para o seu funcionamento de forma dinâmica, também pré-calculando as similaridades entre os atributos.

Não foram identificados trabalhos que explorem plataformas distribuídas de código aberto para o processamento em tempo real, como o *Spark* ou *Apache Storm*. Tais plataformas são capazes de processar milhões de entradas por segundo, utilizando um único computador [Srikanth and Reddy 2016]. No *Apache Storm*, o fluxo de dados é definido utilizando uma estrutura de topologia para gerenciar o que deve ser processado a cada instante.

5. Experimento

Nesta seção, será descrito um experimento inicial com objetivo de avaliar se a ferramenta desenvolvida foi capaz de identificar corretamente os pares duplicados e qual a demanda de pares rotulados para configurar o método. Tal experimento é importante para identificar se a Redblock é capaz de agrupar corretamente os pares e de posteriormente construir os pares candidatos.

Para se realizar o experimento foram utilizadas métricas tradicionais. A Precisão avalia, dos pares recuperados, a taxa de pares que foram corretamente identificados. A Revocação mede a taxa de pares recuperados comparando com o total de pares duplicados que estão presentes na base de dados. Por fim, o F1 combina a precisão e a revocação em uma medida única. Além disso, foi utilizada uma base de dados sintética contendo 10.000

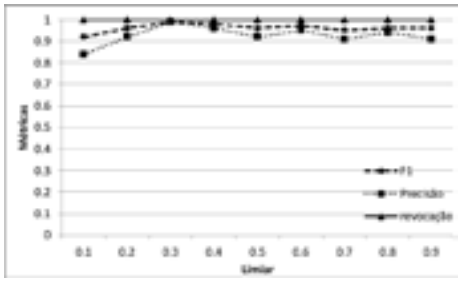

Figura 3. Experimento em uma base de dados sintética com objetivo de avaliar a precisão, revocação e o F1.

registros sendo 1.000 deles duplicatas. A base de dados foi gerada com a ferramenta Febrl [Christen 2008]. Bases sintéticas são importantes devido a possibilidade de controlar o número de pares duplicados e o total de registros que serão inseridos na base de dados.

A Figura 3 apresenta os resultados das métricas F1, precisão e revocação obtidos com a Redblock. O eixo X define o limiar que determina o tamanho do conjunto de treinamento. Por exemplo, o limiar 0.1 define um conjunto de treinamento com 10% de pares positivos (100 pares) e o mesmo número de pares negativos.

Na figura, o limiar 0.1 resultou um valor de F1 de 92%, aumentando para 0.3 o valor é melhorado em 7% atingindo um valor máximo. Percebe-se que o aumento no tamanho do treinamento impacta diretamente na precisão do método. Em outras palavras, quanto mais pares rotulados, mais preciso é o treinamento do método de classificação. É importante notar que a revocação se mantém no valor máximo em todos os limiares, demostrando que a Redblock foi capaz de encontrar todos os pares duplicados da base de dados. Os limiares acima de 0.3 resultaram em uma eficácia bastante instável, ou seja, o aumento no conjunto de treinamento não resultou em uma melhora significativa dos resultados.

Por fim, pode-se notar com essa experimentação inicial que a Redblock foi capaz de promover uma blocagem com alta eficácia, encontrando todos os pares duplicados, assim como o método de classificação, que foi capaz de selecionar boa parte dos pares duplicados[2]

6. Considerações Finais

Neste trabalho, foi proposta uma nova ferramenta para a deduplicação online com foco no processo de blocagem. A ferramenta, denominada *Redblock*, combina o *framework Apache Storm* juntamente com o banco de dados *Redis* para possibilitar um processamento massivo de dados. No experimento inicial, foi possível constatar que a *Redblock* manteve uma alta qualidade (alta eficácia), ou seja, as etapas de blocagem e a classificação foram capazes de recuperar um alto número de pares duplicados sem perdas substanciais de registros positivos.

Os próximos passos envolvem testar a ferramenta com bases de dados contendo milhões de registros para analisar sua eficiência no processamento massivo de dados.

[2]Para visualização contínua dos aprimoramentos no código, o mesmo encontra-se disponível no *Github*. Link para acesso: https://github.com/luanfelixpimentel/storminho

Dessa forma, será possível comparar com outras ferramentas no estado-da-arte da bibliografia. Além disso, será avaliado o desempenho da ferramenta quando aplicada a uma base de dados real, como por exemplo, o *twitter* que é uma ferramenta online e gera uma indeterminada quantidade de dados em tempo real.

Referências

Baxter, R., Christen, P., Churches, T., et al. (2003). A comparison of fast blocking methods for record linkage. In *ACM SIGKDD*, volume 3, pages 25–27. Citeseer.

Bhattacharya, I. and Getoor, L. (2007). Query-time entity resolution. *Journal of Artificial Intelligence Research*, 30:621–657.

Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data-the story so far. *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227.

Christen, P. (2008). Febrl: a freely available record linkage system with a graphical user interface. In *HDKM '08: Proceedings of the second Australasian workshop on Health data and knowledge management*, pages 17–25, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

Christen, P. (2012). A survey of indexing techniques for scalable record linkage and deduplication. *IEEE transactions on knowledge and data engineering*, 24(9):1537–1555.

Dal Bianco, G., Galante, R., Gonçalves, M. A., Canuto, S., and Heuser, C. A. (2015). A practical and effective sampling selection strategy for large scale deduplication. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2305–2319.

Elmagarmid, A. K., Ipeirotis, P. G., and Verykios, V. S. (2007). Duplicate record detection: A survey. *IEEE Transactions on knowledge and data engineering*, 19(1).

Manning, C. D., Raghavan, P., Schütze, H., et al. (2008). *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

Mitra, P., Kang, J., Lee, D., and On, B.-w. (2005). Comparative study of name disambiguation problem using a scalable blocking-based framework. In *Digital Libraries, 2005. JCDL'05. Proceedings of the 5th ACM/IEEE-CS Joint Conference on*, pages 344–353. IEEE.

Ramadan, B., Christen, P., Liang, H., Gayler, R. W., and Hawking, D. (2013). Dynamic similarity-aware inverted indexing for real-time entity resolution. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 47–58. Springer.

Srikanth, B. and Reddy, V. K. (2016). Efficiency of stream processing engines for processing bigdata streams. *Indian Journal of Science and Technology*, 9(14).

Whang, S. E., Menestrina, D., Koutrika, G., Theobald, M., and Garcia-Molina, H. (2009). Entity resolution with iterative blocking. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 219–232. ACM.

Combinando Técnicas de Recomendação e *Smart Posters*

Joedeson Fontana Junior[1], Carlos Vinicius F. Gracioli[1], Daniel Lichtnow[1]

[1]Colégio Politécnico – Universidade Federal de Santa Maria (UFSM)
Av. Roraima nº1000, Campus UFSM – 97.105-900 – Santa Maria – RS – Brasil

jrgfbpa@redes.ufsm.br, carlosviniciusf.gracioli@hotmail.com,
dlichtnow@politecnico.ufsm.br

Abstract. *This paper presents an application that use technologies related to Ubiquitous and Mobile Computing for recommending posters to conference participants. In the work, posters are Smart Posters – posters that has affixed to or embedded in it, one or more readable NFC tags. Together with NFC, techniques used in Recommender Systems have been explored. The application shows more information about a poster that a user is viewing, allows that a user to evaluate the poster, and recommend other posters available at the event, take into account the user location.*

Resumo. *Este artigo apresenta uma aplicação que explora recursos da Computação Ubíqua e Móvel para construção de um sistema que realiza a recomendação de pôsteres aos participantes de uma conferência. Para fazer isto são usados Smart Posters - pôsteres que possuem afixados ou embutidos uma ou mais etiquetas NFC. Junto com NFC, diferentes técnicas utilizadas em Sistemas de Recomendação foram exploradas. A partir disto, a aplicação exibe informações adicionais sobre um pôster que o usuário de um evento consulta, permite que o usuário avalie o pôster e recomenda outros pôsteres disponíveis no evento, considerando a localização do usuário.*

1. Introdução

Sistemas de Recomendação vêm sendo utilizados em aplicações que cobrem distintos domínios (*eCommerce*, bibliotecas digitais, recomendação de filmes, etc.). Atualmente, uma série de recursos relacionados à Computação Ubíqua/Móvel permite vislumbrar novas possibilidades de aplicação das técnicas utilizadas nos Sistemas de Recomendação. Estas novas possibilidades apenas começaram a ser exploradas. A partir desta constatação, o objetivo do presente trabalho é usar recursos relacionados à Computação Ubíqua/Móvel, em um domínio de aplicação específico, que permita demonstrar algumas destas possibilidades.

O domínio de aplicação escolhido é o de sessões de pôsteres em conferências científicas, onde em cada pôster é apresentada uma versão resumida de um artigo científico. Neste cenário, os participantes da conferência vão para a área reservada para a sessão de pôsteres, procurando identificar os trabalhos que são de seu interesse. Ocorre que alguns destes trabalhos podem passar despercebidos por muitos participantes. Neste sentido, seria desejável que os participantes fossem notificados sobre a presença de pôsteres de interesse, o que poderia ser feito mediante o uso das técnicas usadas nos Sistemas de Recomendação.

Atualmente, a recomendação de itens de interesse ocorre normalmente quando o usuário está acessando um sistema na Web. Assim, ao acessar artigos científicos

presentes em uma biblioteca digital na Web, por exemplo, um sistema de recomendação pode recomendar artigos a partir do interesse de um usuário, interesse que poderia ser identificado pela interação com o sistema (e.g. artigos científicos acessados) ou pela avaliação explícita de um artigo (e.g. atribuição de notas). No caso de uma sessão de pôsteres, a dificuldade reside no fato de que o participante não está acessando artigos/itens na Web, mas interagindo com objetos físicos (pôsteres). Assim, existem problemas para identificar os interesses do usuário mediante análise da sua interação com o objeto (e/ou avaliação) e recomendar itens (i.e. pôsteres), e ainda aspectos que em uma biblioteca digital, por exemplo, não são relevantes, como a distância do usuário para os objetos recomendados. Em cenários como este da sessão de pôsteres, em que é preciso identificar a interação dos usuários com objetos físicos, para então gerar recomendações, recursos associados à Computação Ubíqua/Móvel podem ser úteis.

A partir disto, foi desenvolvido um aplicativo para *smartphone* que realiza recomendação de pôsteres para os participantes de uma sessão de posterês. Para tornar a interação do usuário com o pôster possível e identificar quando ela ocorre, a aplicação faz o uso da tecnologia *NFC* (*Near Field Communication*), criando os chamados *Smart Posters*. Basicamente, um *Smart Poster* é um objeto/pôster que possui afixado ou embutido uma ou mais etiquetas *NFC*[1]. Etiquetas *NFC,* estão relacionadas a tecnologia *RFID - Radio-Frequency Identification* [Want 2006], consistindo de pequenas etiquetas que podem ser colocadas em objetos físicos de modo a permitir a troca de dados com dispositivos móveis que tenham suporte a tecnologia. Assim, no aplicativo desenvolvido, quando o usuário aproxima o *smartphone* com a aplicação da etiqueta *NFC* presente no pôster, ele pode receber informações complementares relacionadas ao artigo apresentado no pôster e também recomendações de outros pôsteres.

Na aplicação desenvolvida, a recomendação é produzida por meio de uma abordagem Híbrida que faz uso da abordagem Baseada em Conteúdo, Filtragem Colaborativa e considera a localização dos pôsteres, priorizando aqueles que estão mais próximos do usuário. Embora voltado para um domínio específico, o desenvolvimento desta aplicação pode servir como base para análise de outras possibilidades de integração entre recursos da computação Móvel/Ubíqua, cada dia mais presentes no cotidiano das pessoas, e os Sistemas de Recomendação, que poderiam passar a indicar então objetos físicos.

O artigo inicia na Seção 2, onde são apresentados trabalhos relacionados à área de Sistemas de Recomendação, sendo dado ênfase aqueles sistemas que usam tecnologias que são frequentemente relacionadas a chamada Computação Ubíqua e a Internet das Coisas (especialmente *RFID* e *NFC*). Na Seção 3 é apresentado o aplicativo, sua arquitetura, cenário de uso e as tecnologias utilizadas. Por fim, a Seção 4 apresenta as considerações finais, sendo destacadas as contribuições do trabalho e perspectivas para sua continuidade.

2. Trabalhos Relacionados

Sistemas de Recomendação tem por objetivo identificar itens (artigos, livros, filmes, etc) que possam ser úteis para seus usuários [Adomavicius e Tuzhilin 2005]. Um grande número de autores considera três abordagens básicas para estes sistemas: Baseada em Conteúdo, Colaborativa e Híbrida [Adomavicius e Tuzhilin 2005]. Na abordagem

[1] http://nfc-forum.org/wp-content/uploads/2013/12/NFC-Smart-Poster-WIMA-2011.pdf

Baseada em Conteúdo são recomendados itens que tenham similaridade com itens que o usuário gostou no passado. Na Colaborativa são recomendados itens que foram bem avaliados por pessoas que tenham gosto similar aos do usuário alvo. Já a abordagem Híbrida combina diferentes abordagens, tentando minimizar problemas inerentes a cada uma das diversas abordagens existentes.

Os primeiros Sistemas de Recomendação não levavam em conta o contexto (local, hora, clima, etc.), porém, já há algum tempo, vários sistemas utilizam informações contextuais. A caracterização do contexto e a coleta de dados que permitam sua caracterização são temas fortemente relacionados à Computação Ubíqua que é caracterizada pela integração transparente dos recursos computacionais ao dia a dia das pessoas [Weiser 1991]. Além de relacionada à mobilidade (Computação Móvel), a Computação Ubíqua está relacionada à Computação Pervasiva, que enfatiza o fato dos dispositivos terem a capacidade de obter do ambiente dados que permitam criar modelos computacionais para ajustar o comportamento de aplicações [Araújo 2003]. Relacionada ainda a Computação Ubíqua está a denominada Internet das Coisas - *Internet of Things* - *IoT* [Ashton, 2009] que aborda questões relacionadas a como possibilitar que objetos físicos (carros, refrigeradores, roupas, etc.) estejam conectados a Internet, possuindo um endereço único, podendo adquirir informações sobre seus estados e sobre o ambiente que os cerca, ser monitorados e comunicar-se entre si [Aggarwal et al. 2013].

O cenário proposto pela Computação Ubíqua e *IoT* permite vislumbrar uma série de possibilidades em Sistemas de Recomendação, gerando o que alguns autores denominam Sistemas de Recomendação Ubíquos, que são basicamente aqueles que exploram características dos sistemas ubíquos para gerar recomendações, i.e. sistemas que obtém vantagens dos avanços da telefonia móvel, das conexões *wireless* e da capacidade que dispositivos possuem de obter informações sobre o ambiente onde estão [Mettouris e Papadopoulos, 2014]. Estes sistemas não representam necessariamente uma nova abordagem para Sistemas de Recomendação, podendo utilizar abordagens tradicionais (e.g. Baseada em Conteúdo, Colaborativa, Híbrida). Embora o cenário vislumbrado na Computação Ubíqua e na Internet das Coisas não seja ainda uma realidade plena, já que muitas questões permanecem pendentes de solução, é possível encontrar trabalhos que descrevem sistemas que podem ser considerados Sistemas de Recomendação Ubíquos. Neste sentido, uma análise de trabalhos que envolvem Sistemas de Recomendação e Computação Ubíqua é apresentada em [Rudel; Gubiani; Lichtnow, 2014), sendo alguns destes trabalhos são destacados a seguir.

Em [Walter et al., 2012], por exemplo, é discutido o uso de Sistemas de Recomendação em lojas de varejo, considerando o uso de etiquetas *RFID* (*Radio Frequency IDentification*) nos produtos. A tecnologia *RFID* permite identificar e rastrear objetos por meio de ondas de rádio, sendo apontada como uma das tecnologias básicas para construção da *IoT*. Outro exemplo do uso da tecnologia *RFID* na construção de sistemas de recomendação é apresentada em [Huang et al., 2010] e [Karimi et al., 2012] onde durante a visita a um museu, a interação dos visitantes com os objetos do museu (que possuem etiquetas *RFID*) é acompanhada, sendo feitas recomendações a eles. Usando também etiquetas *RFID* nos objetos, em [Yao et al., 2014] é proposta a recomendação de objetos físicos presentes em um ambiente para seus frequentadores - o sistema procura prever o uso de um objeto (um utensílio de cozinha, por exemplo) a partir da utilização prévia de outro. Já em [Garcia-Perate et al., 2013] é feito um experimento que utiliza a técnica de Filtragem Colaborativa para

recomendar vinhos para os clientes a partir da interação deles com garrafas dispostas em uma mesa (o cliente realiza a leitura de um código de barra presente na garrafa e vinhos são indicados mediante variação das cores dos *RGB Leds* presentes nas garrafas).

Além de trabalhos que fazem uso de etiquetas *RFID*, é possível identificar trabalhos que fazem uso da tecnologia *NFC*, tecnologia presente atualmente em muitos *smartphones*. Em [Luo; Feng, 2015], por exemplo, é descrita a proposta de uma aplicação para *smartphones*, na qual o usuário faz a leitura de *tags NFC* presentes em pôsteres de propaganda de livros em uma livraria, sendo recomendados livros do mesmo gênero daqueles bem avaliados. Neste sistema a recomendação usa técnicas bastante simples (apenas uma taxonomia dos gêneros dos livros é considerada). Outro exemplo do uso da tecnologia *NFC* na criação de *Smart Posters* é apresentado em [Garrido et al., 2010], mas neste trabalho a aplicação apenas apresenta informações que complementam aquelas presentes no pôster. A tecnologia *NFC* tem sido também considerada para estabelecer o relacionamento entre objetos considerando a interação dos usuários [Alves et al., 2015].

O presente trabalho usa técnicas de recomendação e faz uso da tecnologia *NFC,* algo que ainda não foi muito explorado. Em relação aos trabalhos relacionados, no presente trabalho é utilizada uma abordagem Híbrida, que leva ainda em conta a localização do usuário, i.e. a distância do usuário em relação aos objetos recomendados.

3. Sistema de Recomendação para *Smart Posters*

Considerando algumas das possibilidades geradas pelos recursos relacionados à Computação Ubíqua e a Internet das Coisas descritas na Seção 2 e os trabalhos analisados, foi criado um sistema de recomendação onde os itens a serem recomendados pelo sistema consistem de pôsteres cujo conteúdo está relacionado a artigos apresentados em uma conferência. Os pôsteres são *Smart Posters* - pôsteres que possuem afixadas ou embutidas uma ou mais etiquetas *NFC*. A ideia geral é permitir a interação dos participantes das conferências com os objetos (*Smart Posters*), para então gerar recomendações de outros *Smart Posters* para estes participantes. Detalhes sobre aplicação e um cenário de uso são descritos nas próximas seções. A opção pelo uso da tecnologia *NFC* está relacionada ao fato de que ela está presente em um grande número de *smartphones*, e tem um custo e complexidade menor se comparada a soluções que utilizam *RFID*.

3.1. Arquitetura

Basicamente, o sistema desenvolvido consiste em um aplicativo para *smartphones* que utilizam *Android*. O aplicativo permite o cadastro e a autenticação de usuários e uma vez realizado o cadastro e a posterior autenticação, o aplicativo possibilita a leitura de etiquetas *NFC* presentes nos pôsteres, para então mostrar ao usuário informações complementares sobre o pôster de interesse e recomendações.

A arquitetura do aplicativo é mostrada na Figura 1. Existe um repositório de dados da aplicação, construído com *PostgreSQL*, que é acessado usando *Web Services*. O primeiro *Web Service* (na Figura 1 referenciado como *WebService 1*) é responsável por controlar e gerenciar todas as iterações do usuário, seguindo o padrão de projeto *MVC* (*Model*, *View*, *Controller*). As principais funcionalidades controladas por esta *Web Service* são: i) cadastrar e autenticar o usuário, ii) buscar informações

complementares (texto completo e média de avaliações) sobre o pôster de interesse, iii) possibilitar ao usuário avaliar o pôster e/ou solicitar uma recomendação e iv) apresentar os itens recomendados. Já o segundo *Web Service* (na Figura 1 referenciado como *WebService 2*) é responsável pelo processamento da recomendação, usando para isto, recursos existentes no *Apache Mahout* e no *PostgresSQL* (detalhes são descritos na Seção 3.3).

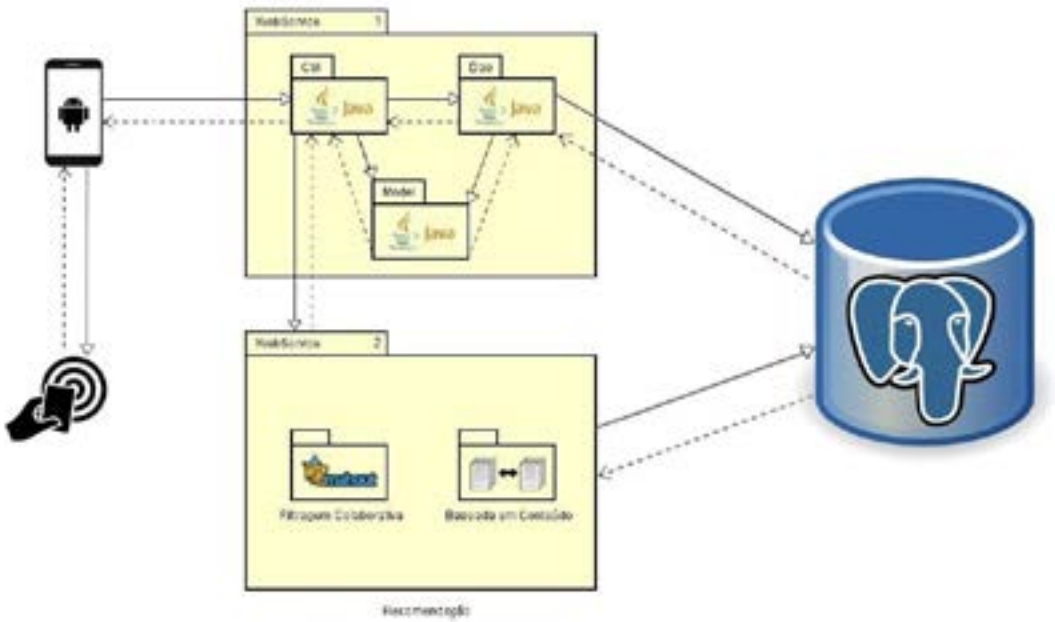

Figura 1. Arquitetura do aplicativo.

No repositório de dados da aplicação são armazenadas informações sobre o usuário (nome, *login, email* e senha), dados sobre os artigos relacionados aos pôsteres apresentados em uma sessão (titulo, texto do artigo, autores, etc.), a localização do pôster no ambiente onde é realizada a sessão de pôsteres da conferência, as notas atribuídas pelos usuários aos pôsteres (ver Seção 3.2), dados sobre a iteração do usuário (e.g. quando o usuário efetua a leitura de uma etiqueta *NFC* esta ação é registrada, quando o usuário acessa uma recomendação realizada esta ação é registrada) e as recomendações geradas para o usuário.

3.2. Cenário de Uso do Sistema

O diagrama de atividades da Figura 2 ilustra o uso e funcionamento do aplicativo. Após a autenticação, o aplicativo apresenta ao usuário a interface mostrada na Figura 3(A), sendo possível ao usuário aproximar o *Smartphone* da etiqueta presente no pôster para efetuar sua leitura.

A etiqueta *NFC* presente no pôster armazena um identificador para o pôster, que uma vez lida com o uso do *smartphone*, apresenta por meio do aplicativo informações complementares sobre o trabalho presente no pôster e permite ao usuário realizar a avaliação, ou visualizar a recomendação conforme indicado na Figura 3 (B).

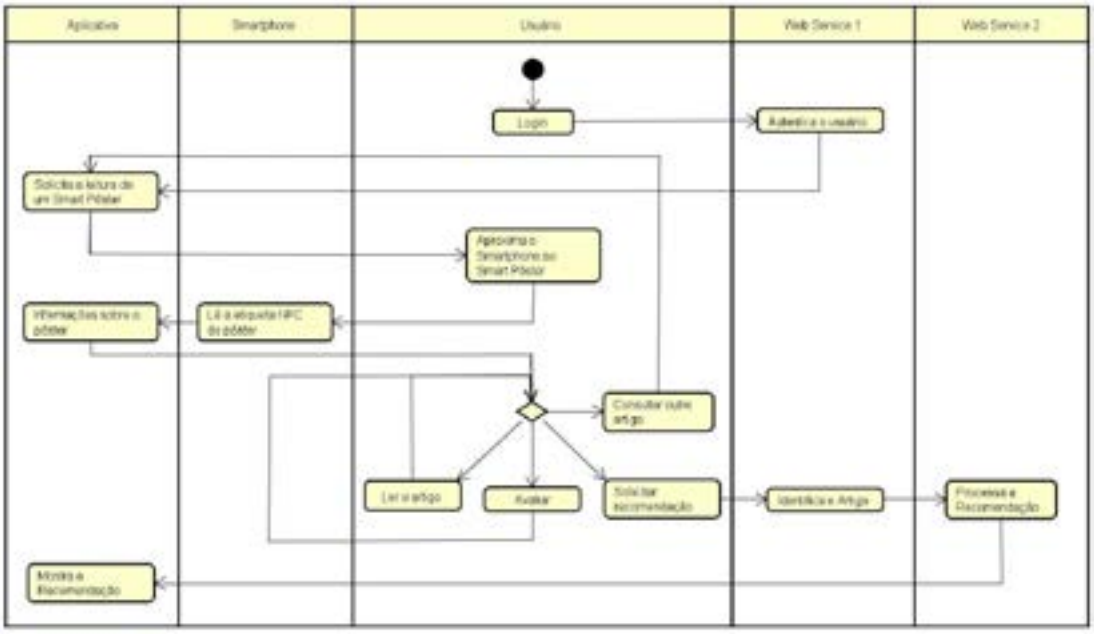

Figura 2. Uso do Aplicativo

Caso o usuário opte pela opção "*ler o texto avaliar*" - Figura 3(B) –, será mostrada a interface da Figura 4(A) permitindo que o usuário faça a leitura do texto, visualize os demais dados do pôster e realize a avaliação. Já caso o usuário opte por receber as recomendações, ele pode escolher a opção *similaridade* - Figura 3(B) – sendo então mostrados pôsteres similares ao que ele está visualizando (i.e. lendo a etiqueta *NFC*) ou a opção *avaliação* que faz uso da Filtragem Colaborativa para gerar a recomendação. Detalhes sobre o processo de recomendação são descritos na Seção 3.3.

Figura 3. Interface do aplicativo para Smart Posters.

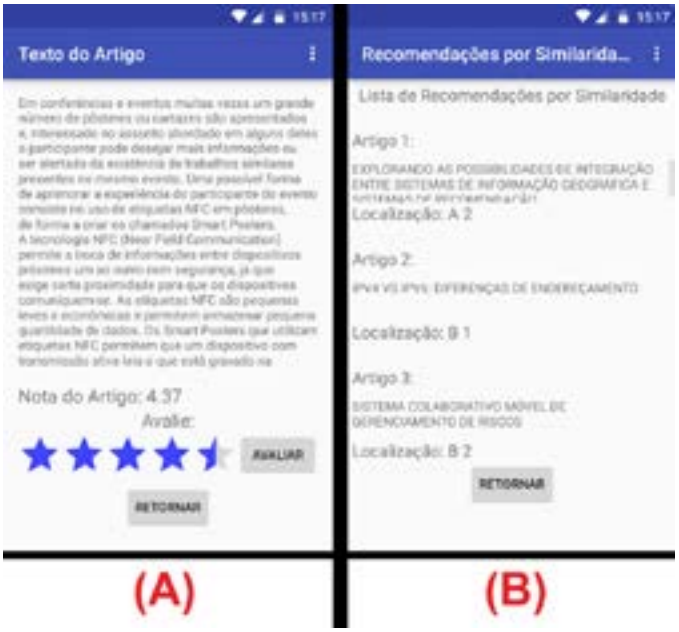

Figura 4. Interface do aplicativo para Smart Posters.

3.3. Processo de Recomendação

A abordagem utilizada para gerar a recomendação é híbrida, combinando as abordagens Baseada em Conteúdo e Colaborativa. Seguindo [Burke, 2002] o método de combinação usado é misto (*mixed*), uma vez que as duas formas de recomendação são apresentadas ao usuário – ver Figura 4(B), cabendo ao usuário escolher entre uma das duas abordagens. Inicialmente, em uma versão preliminar da ferramenta, apenas a abordagem Baseada em Conteúdo era usada [Gracioli e Lichtnow, 2016].

O uso das duas abordagens aqui é justificado pelas já conhecidas limitações das abordagens Baseada em Conteúdo e Colaborativa. Estas limitações referem-se ao problema do novo usuário (*new user* – usuários novos não possuem um perfil formado) e do novo item (*new item* – pôsteres não avaliados não serão recomendado para nenhum usuário) e a questão da ausência de avaliação de qualidade e super especialização (sempre mais do mesmo) presente na abordagem Baseada em Conteúdo.

Seguindo a abordagem Baseada em Conteúdo, o sistema irá recomendar outros pôsteres cujo texto do artigo relacionado é similar em relação ao artigo associado pôster de interesse i.e. o pôster cuja etiqueta *NFC* foi lida pelo usuário. O grau de similaridade é calculado usando recursos presentes no *PostgreSQL,* que é o banco utilizado para armazenar os dados da aplicação. O principal recurso utilizado é a função *similarity*, esta função está disponível no módulo *pg_trgm*[2], que disponibiliza funções e operadores para determinar a similaridade entre textos baseado na comparação de trigramas. Assim, a função *similarity* compara um texto de um pôster com todos os outros armazenados no banco de dados e para cada comparação retorna um valor entre 0 e 1, sendo que quanto mais próximo de 1 maior é a similaridade.

Já na Filtragem Colaborativa serão recomendados pôsteres bem avaliados por usuários com gostos similares ao usuário alvo, levando em consideração os pôsteres avaliados pelos usuários. Na Filtragem Colaborativa, é feito uso dos algoritmos

[2] https://www.postgresql.org/docs/9.6/static/pgtrgm.html

presentes no A*pache Mahout*[3], basicamente é usado o coeficiente de *Pearson* para determinar a similaridade entre usuários. Como o número de usuários e itens não é grande como em alguns domínios (*eCommerce*, por exemplo), o processamento é feito sempre que um usuário indicar que deseja obter o resultado deste tipo de recomendação.

No processo de recomendação são levados em conta aspectos relacionados ao contexto do usuário, mais especificamente sua localização. Cabe ressaltar que esta localização não é obtida através de um sistema de posicionamento global (como coordenadas de um *GPS*, por exemplo), pois um sistema de posicionamento global tende a sofrer interferências quando utilizado em um ambiente interno, propício ao cenário da aplicação atual. Assim, a abordagem assume os pôsteres dispostos em um sistema de coordenadas (uma matriz) onde cada espaço/célula é identificado (Figura 5), sendo calculada a distância entre os pôsteres usando a Distância de *Manhattan,* que determina a distância entre dois pontos $p_1(x_1, y_1)$ e $p_2 (x_2, y_2)$ é calculada conforme demonstrado em (1).

$$| x_1 - x_2 | + | y_1 - y_2 | \qquad (1)$$

A1	**B1**	**C1**
A2	**B2**	**C2**
A3	**B3**	**C3**

Figura 5. Distribuição dos pôsteres.

Assim, as recomendações são apresentadas ao usuário de forma a ordenar os pôsteres do mais próximo para o mais distante em relação ao pôster que ele está interagindo (i.e. o pôster onde o usuário leu a etiqueta *NFC*). Portanto, se o usuário estiver, por exemplo, visualizando detalhes do pôster que está na posição C3 e se fossem recomendados, usando ou a abordagem Baseada em Conteúdo ou a Filtragem Colaborativa, os pôsteres localizados nas posições A1, B1 e C2, a ordem em que a recomendação seria apresentada seria C2, B1 e A1.

4. Considerações Finais

Existem expectativas expressas em alguns trabalhos de que os Sistemas de Recomendação poderão ser úteis no mundo físico como vem sendo no mundo digital, "preenchendo um importante gap na Computação Ubíqua" [McDonald 2003]. A partir disto, este trabalho descreveu um sistema de recomendação que faz uso de abordagens tradicionais de recomendação e de recursos relacionados à Computação Ubíqua/Móvel

[3] https://mahout.apache.org/

para recomendar objetos físicos - pôsteres de uma conferência, dispostos em um ambiente para participantes desta conferência.

Em relação aos trabalhos relacionados, o trabalho faz uso de técnicas de recomendação de uma forma que não foi encontrada nos trabalhos analisados, que se utilizam alguma das técnicas tradicionais de recomendação não consideram a localização do usuário. Cabe destacar a opção pelo uso de *NFC*, hoje presente em um grande número de *smartphones*, o que facilita o desenvolvimento de sistemas para outros domínios que utilizem abordagens similares.

Um dos principais desafios encontrados no desenvolvimento da aplicação é a da localização em um ambiente fechado, como um salão de exposições. O *GPS* (tecnologia disponível na maioria dos *smartphones* atuais) não possui a precisão necessária para determinar localização de cada pôster em um evento (especialmente se realizado dentro de um prédio). Para isso foi utilizada a Distância de Manhattan, dispondo os pôsteres em forma de matriz e informando ao usuário a localização com base na linha e coluna da matriz. Outras soluções podem ser pensadas, mas cabe dizer que a localização *indoor* é um problema ainda carente de solução [Lymberopoulos et al., 2015].

É esperado que a aplicação exibida neste artigo ajude a aprimorar a experiência de um usuário que frequente um evento em que *Smart Posters* são exibidos. Foram feitas avaliações e testes preliminares e pretende-se aprofundar esta avaliação do por meio do Modelo de Aceitação Tecnológica proposto em [Davis, 1989] que observa que dois fatores principais influenciam na aceitação de uma tecnologia: percepção da utilidade (*perceived usefullness*) e facilidade de uso percebida (*perceived ease of use*).

Embora desenvolvido para um domínio específico, é possível pensar em outros cenários de aplicação da abordagem usada no desenvolvimento do sistema. Etiquetas *NFC* podem ser colocadas em outros objetos, dentro de ambientes (quadro de avisos, por exemplo), conforme visto em alguns trabalhos citados na Seção 2. Além disto, dados sobre interação do usuário podem enriquecer o processo de recomendação.

Agradecimentos

Trabalho apoiado pelo Programa de Bolsas de Ensino, Pesquisa e Extensão do Colégio Politécnico da UFSM.

Referências

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, 17(6), 734-749.

Aggarwal, C. C., Ashish, N. and Sheth, A. (2013) The internet of things: A survey from the data-centric perspective, in Managing and mining sensor data, p. 383-428.

Alves, T. M., da Costa, C. A., da Rosa Righi, R., and Barbosa, J. L. V. (2015) Exploring the social Internet of Things concept in a univeristy campus using NFC, in Computing Conference (CLEI), 2015 Latin American, p. 1-12.

Araújo, R. B. (2003). Computação ubíqua: Princípios, tecnologias e desafios. In XXI Simpósio Brasileiro de Redes de Computadores (Vol. 8, pp. 11-13).

Ashton, K. (2009) That ‘internet of things’ thing, RFiD Journal, 22(7), p. 97-114.

Burke, R. (2002) Hybrid recommender systems: Survey and experiments, User modeling and user-adapted interaction, 12(4), p. 331-370.

Davis, F.D. (1989) Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quarterly 13(3), pp. 319-340.

Garcia-Perate, G., Dalton, N., Conroy-Dalton, R., and Wilson, D. (2013) Ambient recommendations in the pop-up shop, in Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing, p. 773-776.

Garrido, P. C., Miraz, G. M., Ruiz, I. L., and Gómez-Nieto, M. Á. (2010) A near field communication tool for building intelligent environment using smart posters, in International Journal of Computers and Communications, 4(1), p. 9-16.

Gracioli, C. V. F.; Lichtnow D. Um sistema de recomendação baseado em conteúdo para smart posters de conferências. in Proceedings of Jornada Acadêmica Integrada – JAI , 2016, p.1.

Huang, Y. P., Chang, Y. T., and Sandnes, F. E. (2010) "Experiences with RFID-based interactive learning in museums", in International Journal of Autonomous and Adaptive Communications Systems, 3(1), p. 59-74.

Karimi, R., Nanopoulos, A., and Schmidt-Thieme, L. (2012) RFID-enhanced museum for interactive experience, in Multimedia for Cultural Heritage, p. 192-205.

Lymberopoulos, D., Liu, J., Yang, X., Choudhury, R. R., Handziski, V., & Sen, S. (2015). A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned. In Proceedings of the 14th international conference on information processing in sensor networks p. 178-189.

Luo, J., and Feng, H. (2015) A Framework for NFC-based Context-aware Applications, in International Journal of Smart Home, 9(1), p. 111-122.

McDonald, D. W. (2003). Ubiquitous recommendation systems. Computer, 36(10), 111-112.

Mettouris, C., Papadopoulos, G. (2014) Ubiquitous recommender systems. Computing. Springer Vienna, p. 1-35.

Rudel, I. E. V., Gubiani, J. S. and Lichtnow, D. (2014) Sistemas de Recomendação e Computação Ubíqua: Um Survey. in Escola Regional de Banco de Dados, 2015, Caxias do Sul-RS. Anais ERBD, 2015, p. 1-10.

Walter, F. E., Battiston, S., Yildirim, M. and Schweitzer, F. (2012) Moving recommender systems from on-line commerce to retail stores., in Information Systems and e-Business Management 10(3), p. 367-393.

Want, R. (2006). An introduction to RFID technology. IEEE pervasive computing, 5(1), 25-33.

Weiser, M. (1991). The computer for the 21st century. Scientific american, 265(3), 94-104.

Yao, L., Sheng, Q. Z., Ngu, A. H., Ashman, H., and Li, X. (2014) Exploring recommendations in internet of things. in Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, p. 855-858.

Inclusão de Técnicas de Interpolação de Pontos em Algoritmos de Descoberta *On-Line* do Padrão *Flock*

Vitor Hugo Bezerra[1], **Daniel dos Santos Kaster**[1]

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

vitorbezera@gmail.com, dskaster@uel.br

Abstract. *With decreasing costs of tracking devices and their availability in vehicles, smartphones and other devices, there is an increase of spatio-temporal data, which can be mined to find patterns regarding groups of moving objects. Among such patterns is the flock pattern that can be defined as a minimal number of entities within a defined distance radius moving together during a certain time-window. However, collection of object positions is usually irregular due to problems such as system failures or passing through tunnels or underground, resulting in gaps in trajectories that may preclude detection of moving patterns. One technique to address this problem is path interpolation, which geometrically generates the missing spatio-temporal points using a given estimation method. The objective of this work is the implementation and evaluation of the inclusion of interpolation techniques for the treatment of entries for on-line flock pattern algorithms. We present experiments that showed good results on finding flock patterns by applying interpolation compared to the results of utilizing the original datasets.*

Resumo. *Com a redução de custo e a maior disponibilidade de dispositivos de localização em veículos, smartphones e outros aparelhos, há um aumento de dados espaço-temporais, que podem ser minerados a fim de se encontrar padrões em grupos de objetos em movimento. Entre esses padrões está o padrão flock, que pode ser definido como um número mínimo de entidades dentro de um espaço delimitado por uma circunferência de raio definido que se deslocam juntos por um certo intervalo de tempo. No entanto, a coleta de posições de objetos é usualmente irregular devido a problemas, como falha de sistema ou falha por passagem em túneis, resultando em perdas nas trajetórias coletadas que podem impedir a identificação de padrões de movimentação. Uma solução para este problema é a interpolação de pontos que calcula geometricamente os pontos faltantes utilizando algum método de estimativa. O objetivo deste trabalho é a implementação e avaliação da inclusão de técnicas de interpolação para o tratamento de entradas para algoritmos de descoberta on-line do padrão flock. São apresentados experimentos que mostraram bons resultados na busca por padrões flock aplicando-se interpolação quando comparados aos resultados utilizando as bases de dados originais.*

1. Introdução

A crescente quantidade de dados espaço-temporais adquiridos atualmente tem ressaltado a necessidade de algoritmos para interpretar esses dados armazenados em grandes bancos de dados. A análise desse tipo de dado complexo, embora seja uma operação cara

computacionalmente, pode identificar comportamentos similares entre objetos como, por exemplo, migração de animais, detecção de congestionamentos em vias e de lugares com grande movimentação. Dentre as várias formas de análise de dados espaço-temporais, estão as de identificação de padrões de grupos de objetos em movimentando, mais especificamente aquele com representação em disco, que explora a distância máxima entre um número mínimo de entidades quaisquer não excedendo um diâmetro de disco definido em uma busca. Um dos padrões mais representativos da categoria de discos é o *flock*, que compreende um conjunto mínimo de objetos que estão espacialmente próximos por um intervalo de tempo pré-definido. Dados um número mínimo de entidades μ, um diâmetro de distância ϵ e um número de instantes de tempo δ, um *flock* é um conjunto de μ ou mais entidades que permanecem por pelo menos δ instantes de tInterpolacaoempo consecutivos respeitando o espaço definido por um disco de diâmetro ϵ em cada instante de tempo [Vieira et al. 2009].

Entretanto, assim como no recebimento quanto na própria coleta desses dados de trajetórias, podem ocorrer falhas e ruídos, acarretando em atrasos e até mesmo perda de certas localizações coletadas. Essa possível irregularidade é causada por problemas como falhas em sistemas e dispositivos, por passagens por túneis ou subterrâneos, pela própria diferenciação das taxas de amostragem dos pontos de localização entre os diversos sistemas de coleta, etc. Como consequência desses possíveis *gaps* nas trajetórias, a identificação dos padrões *flock* pode ficar comprometida, devido à restrição temporal sequencial do padrão. Na prática, isto pode resultar na não identificação de um grupo de animais migrando, um grupo de bandidos praticando assaltos em uma região ou um grupo de turistas se movendo, por exemplo. Para o tratamento das trajetórias problemáticas, uma das técnicas que pode ser utilizada é a interpolação de pontos. Esta consiste em, a partir de determinados pontos coletados, interpolar e inserir pontos nos *gaps* dessas trajetórias.

Neste contexto, o objetivo deste trabalho é tratar o problema de detecção de *flocks* em trajetórias sujeitas a dados com variação de taxa de amostragem e *gaps* por falhas e/ou ruídos na coleta. A proposta é inserir técnicas de interpolação de pontos em algoritmos para a detecção on-line do padrão *flock*, possibilitando o tratamento de *streams* de dados de trajetórias de objetos móveis. Nossa proposta armazena dados recebidos em instantes subsequentes de tempo, guardando-os em estruturas de dados (*buffers*), detecta a falta de uma determinada posição e realiza a interpolação, gerando uma estimada para o objeto. Experimentos realizados em trajetórias com pontos de localização retirados mostram que a proposta obteve uma grande recuperação de resultados perdidos quando comparados com os resultados com as trajetórias originais completas. São reportados resultados que confirmam que a taxa de respostas perdidas foi reduzida em até 80%, embora o uso de interpolação seja sujeito à inclusão de respostas erradas, ou seja, falso-positivos.

O restante deste artigo está organizado da seguinte maneira. A Seção 2 apresenta os conceitos básicos relacionados a este trabalho. A Seção 3 apresenta uma análise de sensibilidade dos algoritmos avaliados frente à falta de dados e a proposta de inclusão da técnica de interpolação. Na Seção 4 são apresentados os resultados obtidos e, por fim, na Seção 5, a conclusão e considerações finais.

2. Fundamentação Teórica

2.1. Padrão *Flock*

Na literatura são encontradas diversas pesquisas a respeito de descoberta de padrões em dados de objetos móveis. Um padrão importante, foco deste trabalho, é o padrão *flock* (*flock pattern*). A definição mais utilizada na literatura define o padrão *flock* como um grupo de pelo menos m entidades movendo-se respeitando uma área máxima de um disco de diâmetro ϵ por um intervalo k de tempo [Vieira et al. 2009, Benkert et al. 2008, Arimura and Takagi 2014, Gudmundsson and van Kreveld 2006, Tanaka et al. 2015]. Formalmente, segundo [Vieira et al. 2009], um *flock* é dado pela Definição 1.

Definição 1 (Padrão *Flock*) *Dado um conjunto de trajetórias $\mathcal{T}$, um número mínimo de trajetórias $\mu > 1$ ($\mu \in \mathbb{N}$) , uma distância máxima $\epsilon > 0$ definida por uma métrica e distância d e uma duração mínima de $\delta > 1$ instantes de tempo ($\delta \in \mathbb{N}$), um padrão **Flock**(μ,ϵ,δ) reporta um conjunto $\mathcal{F}$ contendo todos os flocks f_k, que são conjuntos de trajetórias de tamanho maximal tais que: para cada $f_k \in \mathcal{F}$, o número de trajetórias é maior ou igual a μ e existem δ instâncias de tempo consecutivas t_j, ..., $t_{j+\delta-1}$ em que existe um disco com centro $c_k^{t_i}$ e diâmetro ϵ cobrindo todos os pontos das trajetórias de $f_k^{t_i}$, que é flock f_k no instante t_i, $j \leq i \leq j + \delta$.*

Dentre os algoritmos para a identificação do padrão *flock*, dois merecem destaque por permitirem a detecção *on-line* do padrão: o BFE e o PSI. O algoritmo BFE (***B**asic **F**lock **E**valuation*) foi proposto por [Vieira et al. 2009] e utiliza um índice baseado em grade para processar os dados das trajetórias em cada instante de tempo. Já o algoritmo PSI (***P**lane Sweeping, Binary **S**ignatures and Inverted **I**ndex*), baseado no BFE e proposto por [Tanaka et al. 2015, Tanaka 2016], utiliza técnicas e estruturas de dados para detectar *flocks* de forma mais rápida que o BFE, como varredura de plano ao invés do índice de grade, assinatura binária e índice invertido. Nos testes apresentados pelos autores, o PSI apresentou desempenho superior ao BFE em vários conjuntos de dados reais para a maior parte das combinações de parâmetros do padrão. O mesmo aconteceu para a maior parte das variações dos algoritmos – heurísticas –, cujos detalhes podem ser encontrados nas referências originais ou em [Tanaka 2016].

2.2. Técnicas de Interpolação

Dados de trajetórias de objetos móveis podem ser incertos e incompletos, ou seja, imprecisos por vários motivos. Exemplos vão desde imprecisões na coleta dos pontos da trajetória, pois os dispositivos utilizados podem ser imprecisos, até a dificuldade de captura de dados de um objeto que está se movendo constantemente, já que sua posição só possa ser guardada em um certo período de tempo caso não haja a sincronização no tempo limite permitido [Zheng and Zhou 2011, Parent et al. 2013].

Na literatura há várias técnicas para o tratamento desses dados, incluindo a utilização de técnicas de *range queries* para conseguir os dados ideais para serem utilizados [Zheng and Zhou 2011], o que necessita uso de uma aplicação de banco de dados; aumentar o tempo entre um ponto e outro retirando alguns pontos, criando uma grande alteração no conjunto de dados pela falta de alguns pontos; utilização de filtros como o *Kalman Filter* [Gustafsson et al. 2002], que necessitam guardar uma grande quantidade de dados para realizar um funcionamento com bons resultados e outras. As técnicas

abordadas neste trabalho para tratamento de incertezas realizam interpolação de pontos. Interpolação é a ação de estimar um novo ponto em um instante de tempo não contido na amostra de dados – *gap* na trajetória – com base nos dados existentes na amostra [Li and Revesz 2002]. Para a sua utilização são necessários pontos de um objeto, anteriores e/ou posteriores ao instante de tempo desejado, i.e., faltante, estimando o posicionamento do objeto móvel no instante por meio de equações matemáticas.

A escolha dos métodos concentrou-se em abordagens aplicáveis a dados "brutos" de trajetória, isto é, sem considerar informação adicional, como malha viária e informações semânticas. São eles:

- **Interpolação Linear**: A interpolação linear calcula a localização estimada de um objeto por uma linha reta entre dois pontos reais coletados. Conhecido por ser um método simples de ser implementado, possui resultados muito bons para faltas sensíveis de dados, além de ser utilizado recorrentemente na literatura como parâmetro de comparação em estudos.
- **Interpolação por *Random Walk* Restrito**: A interpolação por *Random Walk* restrito (*constrained random walk*) consiste em realizar uma interpolação independente da posição anterior do objeto móvel na trajetória. A interpolação é criada a partir de amostras aleatórias de duas distribuições: a distribuição do comprimento de passo (l) e a distribuição do ângulo de rotação (θ). Recomendado em trajetórias em que o objeto observado tem um caminho difícil de se prever, como por exemplo, animais que não têm um trajeto usual, pois não seguem um caminho direto até uma localização, como, por exemplo, macacos [Wentz et al. 2003].
- **Interpolação pela Curva de Bézier**: Já a interpolação com curva cúbica de Bézier requer a definição de quatro pontos âncora. Método bastante recomendado para trajetórias cheias de curvas, e.g., animais marítimos [Tremblay et al. 2006].

3. Inclusão de Interpolação em Algoritmos *On-line* do Padrão *Flock*

Para os algoritmos de *flock*, incertezas nos dados de entrada são bastante prejudiciais. Dentre essas, destaca-se a perda de pontos de localização coletados, sendo mais impactante aos algoritmos *on-line*. Essa lacuna, ou *gap*, na trajetória sendo processada em determinado instante de tempo acarreta na eliminação desta em qualquer possível candidato a *flock* durante a janela temporal sendo processada, visto a restrição de consecutividade temporal dada na definição de *Flock* 1. Dessa forma, como impacto, os algoritmos podem identificar *flocks* com menos objetos que o na realidade, devido ao descarte destes com trajetórias com *gaps*, ou até mesmo deixar de reportar outros *flocks*, caso a remoção desses objetos descarte possível candidatos por terem quantidade insuficiente de objetos para formar o padrão. Qualquer algoritmo de identificação de padrões *flock*, sem o devido tratamento, é sensível a esse erro, em especial os *on-line*, incluindo o BFE e o PSI, utilizados neste trabalho.

Esta seção apresenta uma análise do comportamento desses algoritmos *on-line* de detecção do padrão *flock* quando as trajetórias dos objetos em estudo não têm pontos reportados em todos os seus instantes de tempo. Em seguida, é apresentada a proposta de inclusão de técnicas de interpolação nesses algoritmos, bem como os resultados de experimentos que confirmam a eficácia da proposta.

3.1. Análise da Sensibilidade dos Algoritmos do Padrão *Flock* à Falta de Dados

Conforme abordado anteriormente, é sabido que a falta de pontos em certos intervalos de tempo pode impactar significativamente nos resultados obtidos pelos algoritmos de detecção do padrão *flock*. Contudo, nenhum trabalho havia analisado quantitativamente o impacto da falta de pontos nessas respostas. Desta forma, a seguir é apresentada uma análise a respeito desse aspecto, confirmando que de fato os algoritmos são muito sensíveis à falta de dados.

Para simular a falta de pontos em um conjunto de dados e ainda permitir a comparação das respostas obtidas mediante diferentes situações de falta de dados, optou-se por fazer retirada controlada de pontos do conjunto de dados em teste. Desta forma, as respostas obtidas utilizando-se o conjunto de dados original formam a "regra ouro" e os conjuntos com pontos retirados simulam situações de falta de dados, possibilitando avaliar a sensibilidade dos algoritmos quanto a esse quesito. O método de retirada controlada de pontos foi desenvolvido em Python e trata separadamente as trajetórias de cada objeto do conjunto de dados. Para cada trajetória, é feita a contagem dos pontos seguida da retirada aleatória de uma certa porcentagem dos pontos desta, fornecida como parâmetro. O intuito dessa abordagem é garantir que a retirada de pontos seja homogênea para todos os objetos do conjunto de dados, tratando a falta de pontos como um comportamento inerente ao processo de captura do posicionamento dos objetos, e não específico de determinados objetos.

Para a comparação das saídas dos algoritmos, o método desenvolvido inicialmente faz uma limpeza dos dados e guarda os *flocks* reportados (conjunto de objetos que formam cada *flock* e intervalo de tempo) em listas ordenadas pelo instante de início do intervalo de tempo em que o *flock* foi identificado. Em seguida, é feita a comparação dos *flocks* reportados em cada arquivo, ou seja, em cada execução com percentual específico de retirada de pontos, indicando as taxas de *falso-negativos* (i.e., *flocks* que deveriam ter sido reportados, mas não foram) e de *falso-positivos* (i.e., *flocks* que não deveriam ter sido reportados, mas foram), para cada combinação de valores dos parâmetros dos algoritmos de detecção de *flocks* (μ, ϵ e δ).

A análise utilizou os conjuntos de dados originais dos trabalhos que propuseram os algoritmos BFE e PSI, bem como suas variações (vide Seção 2.1). A seguir são apresentados os resultados obtidos sobre os conjuntos de dados *Buses* e *Cars*. *Buses* contém 66.096 posições de ônibus se movendo na cidade de Atenas, Grécia. Já o conjunto *Cars* é composto de 134.264 posições coletadas a partir da movimentação de 183 carros privados movimentando-se em Copenhague, Dinamarca[1]. De cada conjunto de dados foram derivados três outros conjuntos, considerando as porcentagens de retirada de pontos 1%, 5% e 10%. Para melhor avaliar o impacto da retirada de pontos das trajetórias nos resultados, foram executadas 30 repetições de cada conjunto de dados com cada taxa de retirada de pontos (1%, 5% e 10%), sempre retirando-se pontos de forma aleatória em cada repetição.

Conforme esperado, os resultados obtidos pelo BFE e o PSI foram idênticos em todos os casos, portanto, não será feita referência ao algoritmo na análise dos resultados que segue, apenas a qual conjunto de dados. A Figura 1 mostra que os algoritmos são muito sensíveis a faltas de pontos, pois mesmo a falta de 1% dos pontos faz com que

[1] www.daisy.aau.dk

os algoritmos deixem de reportar entre 20 e 60% das respostas esperadas. A retirada de 5% e 10% dos pontos de cada trajetória faz com que praticamente todas as respostas dos algoritmos sejam perdidas. Percebe-se, ainda, que com o crescimento das variáveis μ e δ, há um aumento no número de respostas que são perdidas, pois aumentam a probabilidade de um ponto faltante dissolver um *flock* inteiro. Já o parâmetro ϵ apresentou uma variação menos acentuada de perda de respostas com o crescimento do valor do parâmetro.

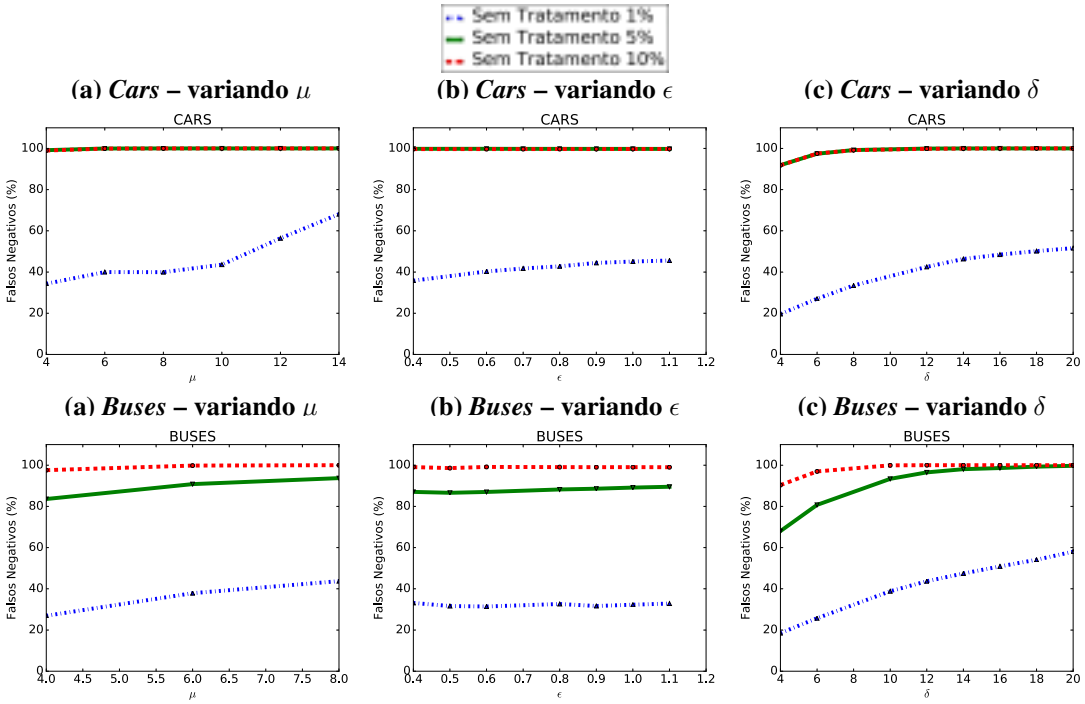

Figura 1. Taxa de falso-negativos gerado pelos algoritmos BFE e PSI mediante diferentes taxas de perdas/retirada de pontos. *Cars* (1ª linha) e *Buses* (2ª linha) variando μ (cardinalidade – 1ª coluna), ϵ (diâmetro dos discos – 2ª coluna) e δ (duração – 3ª coluna).

3.2. Proposta de Inclusão de Interpolação nos Algoritmos *On-line* de *Flocks*

Como visto na seção anterior, a falta do posicionamento de um objeto móvel em um único instante de tempo pode impedir sua inclusão em um *flock*. Isto porque os algoritmos de detecção do padrão *flock on-line* recebem os dados da posição dos objetos a cada instante de tempo. Para contornar essa limitação, a abordagem adotada neste trabalho foi utilizar algoritmos de interpolação de pontos para tornar os algoritmos mais robustos com relação a esse aspecto. Para isso, técnicas de interpolação foram implementadas dentro dos algoritmos BFE e PSI, pois os algoritmos disponíveis de interpolação de uma forma geral consideram que todo o conjunto de dados está disponível, premissa que não é válida para abordagens *on-line*.

O método proposto armazena as posições dos objetos em dados instantes de tempo em estruturas de dados, chamadas aqui de *buffers*. Cada *buffer* guarda as posições de todos os objetos em um certo tempo t, formando uma espécie de janela de tempo da movimentação das trajetórias. A janela é centralizada no *buffer atual*, que é o instante de tempo que poderá receber um ponto interpolado. Os demais *buffers* são utilizados para:

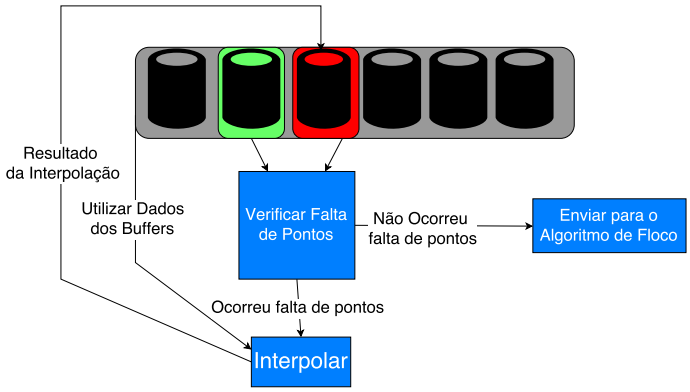

Figura 2. Exemplo do método proposto com a utilização de seis *buffers*.

(a) **identificar a perda de uma posição**: neste aspecto, assume-se que se uma trajetória continha pontos antes e depois do instante de tempo atual, então a falta do ponto no instante atual foi decorrência de uma falha de captura/transmissão; e
(b) **gerar um ponto interpolado**: os pontos nos instantes precedente(s) e subsequente(s) é(são) utilizado(s) para estimar o posicionamento do objeto no instante de tempo em que ocorreu a falha.

Desses *buffers* um é escolhido e os objetos contidos no *buffer* atual são comparados com os do *buffer* $t-1$. Se houver a falta de um objeto, então ele é interpolado com base nos dados desse objeto nos *buffers* de tempos anteriores e/ou futuros. Após a análise deste *buffer*, ele é liberado e os dados seguem para o algoritmo de detecção de *flock*. Se ao tentar interpolar um objeto ele não estiver em nenhum *buffer* futuro, não será interpolado o ponto, o que também funciona como método de parada para a interpolação, assim não ocorrendo interpolações depois do último tempo em que o objeto foi reportado.

Um exemplo do método pode ser visto na Figura 2, em que é ilustrado o método proposto com a utilização de seis *buffers*, representados na figura pelos cilindros. Na imagem, o *buffer* em vermelho é o que está sendo analisado e contém as posições dos objetos em um tempo t. As posições neste *buffer* são comparadas com as posições apresentadas no *buffer* anterior (verde), que contém as posições do instante de tempo $t-1$. Se houver um objeto que reportou uma posição no tempo anterior e não reportou no tempo t, o ponto será interpolado com base nas posições do objeto contidos nos demais *buffers*.

O número de pontos de controle para realizar a interpolação depende do método utilizado. Por exemplo, a interpolação linear e a interpolação por *random walk* restrito precisam de um ponto anterior ao ponto a ser interpolado e um ponto posterior. Já a interpolação pela curva de Bézier precisa de dois pontos anteriores e dois posteriores. Por esse motivo, na figura, há dois *buffers* anteriores ao *buffer* atual. Contudo, o número de *buffers* posteriores é maior devido ao fato que pode ter sido perdida a posição de um objeto em um instante (atual) e também em algum instante logo em seguida (por exemplo, no instante $t+2$). Este tipo de situação inviabilizaria a possibilidade de interpolação pela curva de Bézier. Depois de interpolar as posições necessárias, o *buffer* atual é liberado para o algoritmo de detecção do padrão *flock* processar o instante correspondente.

Vale ressaltar que a abordagem proposta insere um atraso na identificação do padrão *flock*, que depende do número de *buffers* posteriores ao instante atual. No en-

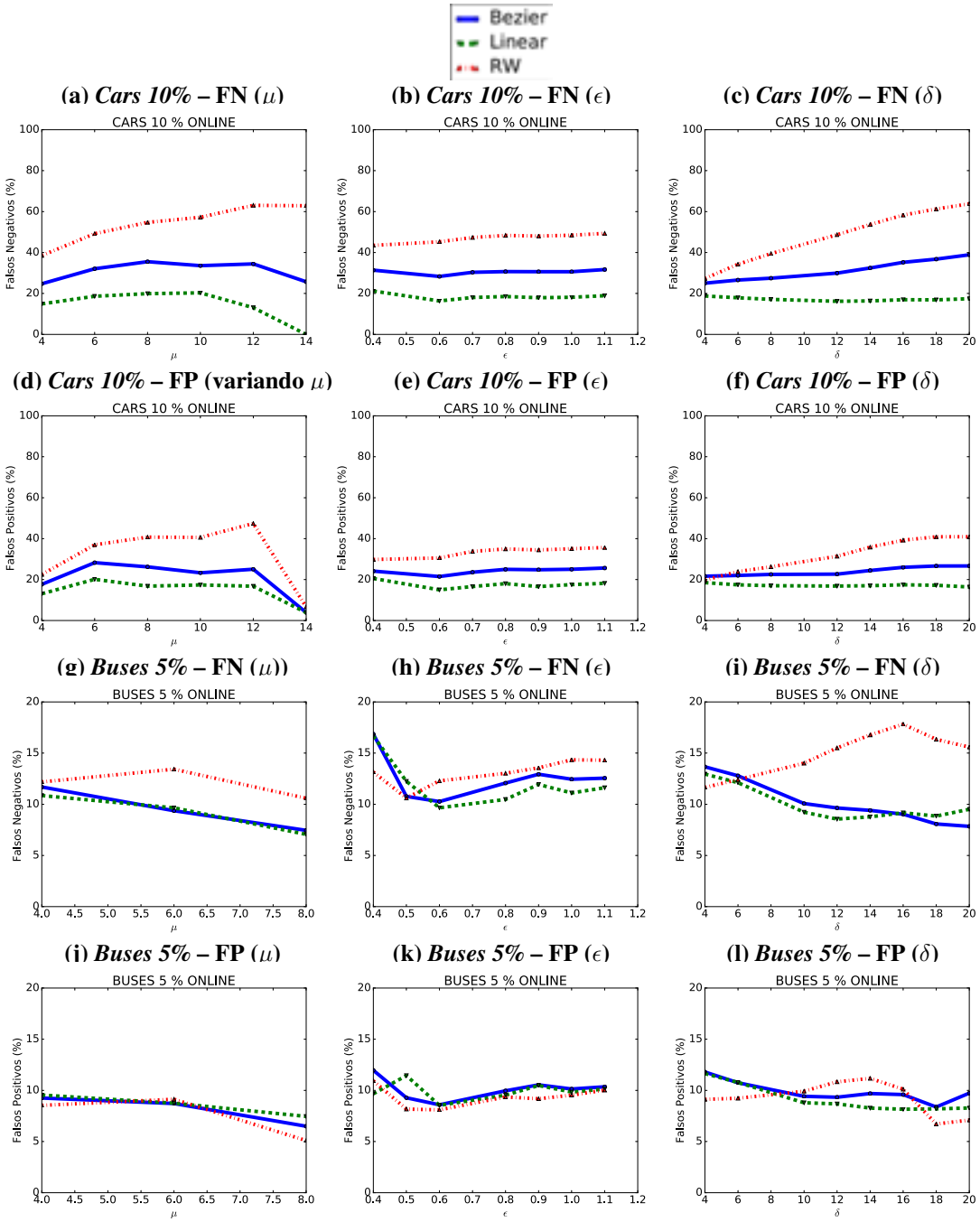

Figura 3. Taxa de acerto do método de interpolação *on-line*, variando μ (cardinalidade – 1ª coluna), ϵ (diâmetro dos discos – 2ª coluna) e δ (duração – 3ª coluna). *Cars* com 10% de perda – falso-negativos (1ª linha) e falso-positivos (2ª linha). *Buses* com 5% de perda – falso-negativos (3ª linha) e falso-positivos (4ª linha).

tanto, considera-se nesta proposta que o atraso é pequeno o suficiente para a maioria das aplicações que dependem da identificação *on-line* de *flocks*. Nestes casos, os benefícios do ganho de qualidade nos resultados sobrepõem o atraso inserido.

4. Avaliação Experimental da Proposta

Esta seção descreve os resultados obtidos sobre os conjuntos de dados *Cars* e *Buses*, avaliando a eficácia da proposta. Foram realizadas baterias de experimentos seguindo o mesmo método de teste e os mesmos parâmetros utilizados na avaliação de sensibilidade dos algoritmos *on-line* do padrão *flock* (Seção 3.1).

A Figura 3 mostra as taxas de acerto da proposta considerando 10% de perda no conjunto *Cars* e 5% no conjunto *Buses* para três técnicas de interpolação implementadas: linear, *random walk* restrito e curva de Bézier. Nota-se que a taxa de falso-negativos caiu consideravelmente em relação à situação em que não foi realizada interpolação para os dois conjuntos (figuras 3(a–c) e 3(g–i)), com destaque para a interpolação linear.

Esta inclusive possibilitou a redução da taxa de falso-negativos de quase 100% no conjunto *Cars* para pouco menos de 20% e a taxa do conjunto *Buses* de mais de 80% para menos de 10%. A curva de Bézier teve resultados pouco menos precisos, parte devido à natureza dos conjuntos de dados considerados, baseados em malha viária. Já a interpolação por *random walk* restrito apresentou o pior desempenho dentre as técnicas, em todos os casos, pois sua interpolação interfere muito na localização dos objetos criando *flocks* diferentes ou retirando objetos de respostas existentes . *Flocks* com menor tempo de duração e maior quantidade de objetos foram os que tiveram melhores recuperações de dados, já a diferença do tamanho do diâmetro do *flock* não resultou em grandes diferenças.

Observe-se que o uso de interpolação é sujeito à geração de falso-positivos, isto é, *flocks* que não existem de fato segundo a regra ouro, mas foram detectados após a inclusão dos pontos interpolados que diferem dos pontos de localização originais. Os gráficos das figuras 3(d–f) e 3(j–l) mostram que os falso-positivos fizeram a taxa total de erros dobrar, na maioria dos casos. Mas há casos em que os falsos positivos são respostas em que se diferenciam das originais pela falta ou adição de um elemento, o que também poderia ser considerado um floco original, o que uma análise menos sensível de diferença de respostas poderia mostrar. Entretanto, ainda usando uma análise sensível a diferenças, a taxa de erro global ainda é consideravelmente inferior do que a que obteve-se sem interpolação, sustentando assim a eficácia da proposta.

5. Conclusão

Na coleta dos pontos de localizações desses objetos móveis, por inúmeros fatores, falhas e ruídos podem ocorrer, como na perda de certas localizações, que, como avaliado neste trabalho, afetam consideravelmente os algoritmos de detecção de padrão *flock*, em especial os *on-line*. Para abordar esse problema, este trabalho avaliou e desenvolveu formas para o tratamento de falta de pontos em trajetórias por meio de técnicas de interpolação de pontos. Foram testados três tipos de interpolação: Linear, *Constrained Random Walk* e Bezier, por obterem melhores resultados de acordo com a literatura. Foi desenvolvido um método de interpolação *on-line*, focado no recebimento de *strems* de dados. Nessa estratégia, como há um número limitado de dados para se realizar as interpolações, também são utilizados os pontos interpolados para a criação de novos pontos.

Também foram realizados diversos experimentos para avaliar o impacto da perda de dados de posicionamento e a eficácia da proposta. Os resultados mostraram que os algoritmos são muito sensíveis visto que uma pequena perda nos pontos já impacta significativamente. Observou-se, também, que quando as variáveis do padrão *flock* são muito

específicas, ou seja, valores pequenos de quantidade de objetos, intervalo de tempo e raio do círculo, a perda de *flocks* tende a ser menor. Os resultados obtidos com a aplicação da proposta possibilitaram um aumento expressivo na qualidade dos resultados, em particular quando foi utilizada a interpolação linear. Trabalhos futuros incluem a aplicação de outras técnicas de interpolação, utilizando técnicas que não dependam de dados futuros, e a avaliação de outras variáveis, como o impacto da proposta no tempo total de execução dos algoritmos e no consumo de memória.

Referências

[Arimura and Takagi 2014] Arimura, H. and Takagi, T. (2014). Finding All Maximal Duration Flock Patterns in High-dimensional Trajectories. *Manuscript, DCS, IST, Hokkaido University, Apr*.

[Benkert et al. 2008] Benkert, M., Gudmundsson, J., Hübner, F., and Wolle, T. (2008). Reporting flock patterns. volume 41, pages 111–125. Elsevier.

[Gudmundsson and van Kreveld 2006] Gudmundsson, J. and van Kreveld, M. (2006). Computing longest duration flocks in trajectory data. In *Proceedings of the 14th ACM GIS*, page 35, New York, New York, USA. ACM Press.

[Gustafsson et al. 2002] Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P.-J. (2002). Particle filters for positioning, navigation, and tracking. *IEEE Transactions on signal processing*, 50(2):425–437.

[Li and Revesz 2002] Li, L. and Revesz, P. (2002). A comparison of spatio-temporal interpolation methods. In *Geographic Information Science*, pages 145–160. Springer.

[Parent et al. 2013] Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., Damiani, M. L., Gkoulalas-Divanis, A., Macedo, J., Pelekis, N., Theodoridis, Y., and Yan, Z. (2013). Semantic trajectories modeling and analysis. *ACM Computing Surveys*, 45(4):42:1–42:32.

[Tanaka 2016] Tanaka, P. S. (2016). *Algoritmos eficientes para detecção do padrão Floco em banco de dados de trajetórias*. Mestrado, Universidade Estadual de Londrina.

[Tanaka et al. 2015] Tanaka, P. S., Vieira, M. R., and Kaster, D. S. (2015). Efficient algorithms to discover flock patterns in trajectories. In *XVI Brazilian Symposium on Geoinformatics (GEOINFO)*, volume 1, pages 56–67, São Paulo, Brazil.

[Tremblay et al. 2006] Tremblay, Y., Shaffer, S. A., Fowler, S. L., Kuhn, C. E., McDonald, B. I., Weise, M. J., Bost, C.-A., Weimerskirch, H., Crocker, D. E., Goebel, M. E., and Others (2006). Interpolation of animal tracking data in a fluid environment. *Journal of Experimental Biology*, 209(1):128–140.

[Vieira et al. 2009] Vieira, M. R., Bakalov, P., and Tsotras, V. J. (2009). On-line discovery of flock patterns in spatio-temporal data. In *Proceedings of the 17th ACM GIS*, pages 286–295, New York, New York, USA. ACM Press.

[Wentz et al. 2003] Wentz, E. A., Campbell, A. F., and Houston, R. (2003). A comparison of two methods to create tracks of moving objects: linear weighted distance and constrained random walk. *Int. Journal of Geographical Information Science*, 17(7):623–645.

[Zheng and Zhou 2011] Zheng, Y. and Zhou, X. (2011). *Computing with spatial trajectories*. Springer Science & Business Media.

Estudo Comparativo de Banco de Dados Chave-Valor com Armazenamento em Memória

Dinei A. Rockenbach[1], **Nadine Anderle**[1], **Dalvan Griebler**[1,2], **Samuel Souza**[1]

[1] Laboratório de Pesquisas Avançadas para Computação em Nuvem (LARCC)
Faculdade Três de Maio (SETREM) – Três de Maio – RS – Brasil

[2]Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS/PPGCC)
Porto Alegre – RS – Brasil

{dineiar,nadianderle}@gmail.com,dalvan.griebler@acad.pucrs.br,
samuel@samuelsouza.com

Abstract. *Key-value databases emerge to address relational databases' limitations and with the increasing capacity of RAM memory it is possible to offer greater performance and versatility in data storage and processing. The objective is to perform a comparative study of key-value databases with memory storage Redis, Memcached, Voldemort, Aerospike, Hazelcast and Riak KV. Thus, the work contributed to an analysis of different databases and with results that qualitatively demonstrated the characteristics and pointed out the main advantages.*

Resumo. *Bancos de Dados (BD) chave-valor surgem para suprir limitações de BDs relacionais e com o aumento da capacidade das memórias RAM é possível oferecer maior desempenho e versatilidade no armazenamento e processamento dos dados. O objetivo é realizar um estudo comparativo dos BDs chave-valor com armazenamento em memória Redis, Memcached, Voldemort, Aerospike, Hazelcast e Riak KV. Assim, o trabalho contribuiu para uma análise de diferentes BDs e com resultados que demonstraram qualitativamente as características e apontaram as principais vantagens.*

1. Introdução

As necessidades de alta disponibilidade e velocidade, bem como o aumento e consumo de dados nos sistemas e aplicações atuais, influenciaram no desenvolvimento de novas formas para o armazenamento de dados. Estas vão além dos bancos de dados relacionais, impulsionando o movimento NoSQL (*Not only SQL*) [Fowler 2015]. Não obstante, com a popularidade em alta em um mercado sem um líder estabelecido, há uma consequente explosão no número de sistemas de armazenamento disponíveis, o que dificulta a tomada de decisão quanto à opção que melhor supre as necessidades organizacionais.

Com o objetivo de solucionar problemas específicos, os bancos NoSQL foram categorizados de acordo com suas características e otimizações. Por exemplo, a Amazon utiliza seu sistema chave-valor (*key-value*) Dynamo [DeCandia et al. 2007] para gerenciar as listas de mais vendidos, carrinhos de compras, preferências do consumidor, gerenciamento de produtos, entre outras aplicações. Exitem também sistemas de família de colunas (*column family* ou *columnar*) que foram influenciados pelo Bigtable da Google [Chang et al. 2008]. Enquanto isso, os assim chamados de sistemas de documentos (*document*) resultaram, por exemplo, no MongoDB[1]. Por fim, do chamado banco triplo

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

(*graph database* ou *triple*), tem-se como exemplo o Neo4j[2]. Cada uma destas categorias traz sistemas que cobrem diferentes limitações dos bancos relacionais tradicionais.

O foco deste trabalho está nos bancos de dados chave-valor com armazenamento em memória, a fim de realizar um estudo comparativo e qualitativo de banco de dados ainda pouco estudados na literatura (Seção 2). Este tipo de banco de dados NoSQL é o representante com as estruturas mais simples dentre os existentes [Pokorny 2013]. No entanto, cabe ressaltar que eles possuem um amplo espectro de casos de uso, sendo largamente adotados tanto como armazenamento primário quanto para auxiliar outros sistemas de armazenamento [Carlson 2013]. Ainda, o aumento na capacidade das memórias RAM fez com estes sistemas se adaptassem para efetuar o armazenamento e processamento de dados em memória, com a finalidade de melhorar o desempenho [Zhang et al. 2015]. A realidade corrobora a afirmação de Jim Gray de que memória é o novo disco e o disco é a nova fita [Robbins 2008], o que justifica o aumento da popularidade destes sistemas.

Este artigo está organizado em 5 seções, incluindo esta seção introdutória. Na Seção 2 estão os trabalhos relacionados. A Seção 3 traz o embasamento sobre os bancos de dados pesquisados. Na Seção 4 está detalhado o estudo comparativo destes sistemas. Por fim, na Seção 5 estão as conclusões e propostas para trabalhos futuros.

2. Trabalhos Relacionados

Nesta seção é apresentada uma discussão sobre os trabalhos relacionados publicados recentemente na literatura. De forma semelhante, todos os trabalhos escolhidos buscam caracterizar e comparar bancos de dados NoSQL. No entanto, não voltam os estudos para um determinado tipo de banco de dados, o que é importante para avaliar características específicas. Tanto [Hecht and Jablonski 2011] quanto [Han et al. 2011] se propõem a fazer um estudo e avaliação dos bancos de dados NoSQL, com o mesmo objetivo principal: prover informações para auxiliar na escolha do banco NoSQL que melhor atende às necessidades. Por não delimitarem um tipo específico de banco NoSQL, ambos possuem um escopo mais abrangente do que o presente trabalho. No ponto de intersecção entre este trabalho e os citados, [Hecht and Jablonski 2011] inclui em seu trabalho os bancos Project Voldemort, Redis e Membase, enquanto que [Han et al. 2011] avalia Redis, Tokyo Cabinet-Tokyo Tyrant e Flare.

Em [Deka 2014] é apresentada uma visão geral de vários sistemas NoSQL e os representantes chave-valor incluídos na avaliação são Hypertable, Voldemort, Dynomite, Redis e Dynamo. Nota-se a falta, porém, de uma visão comparativa mais clara sobre aspectos de garantias de durabilidade, disponibilidade, protocolos suportados, e outras informações que podem vir a ter uma influência significativa na escolha de um banco de dados chave-valor. Já [Zhang et al. 2015] traz uma visão bem estruturada dos objetivos que nortearam o projeto de cada um dos sistemas descritos no trabalho, o qual foca em sistemas com gerenciamento e processamento de dados em memória. Dentre os sistemas estudados, os representantes dos bancos chave-valor são MemepiC, RAMCloud, Redis, Memcached, MemC3 e TxCache. Dos sistemas, o trabalho descreve as cargas de dados mais adequadas ao sistema, a estratégia para construção de índices, o controle de concorrência, tolerância a falhas, tratamentos para conjuntos de dados maiores do que a memória disponível e o suporte a consultas personalizadas em baixo nível (como *stored procedures* e *scripts* em linguagem nativa, por exemplo), porém com pouca abordagem de alto nível que auxilie na escolha de um sistema em favor de outro.

[2]https://neo4j.com

Ainda que o tema NoSQL tenha sido bastante explorado na academia, e a bibliografia focada no armazenamento de dados em memória tenha crescido muito nos últimos anos, nota-se a falta de um estudo comparativo entre os sistemas chave-valor em memória Redis, Memcached, Voldemort, Aerospike, Hazelcast e Riak KV.

3. Banco de Dados com Armazenamento em Memória

O crescimento dos bancos de dados com armazenamento de dados em memória (IMDB ou *in-memory databases*) segue uma tendência que teve seu início no hardware, com a capacidade da memória dobrando em média a cada três anos e seu preço caindo uma casa decimal a cada cinco anos [Zhang et al. 2015]. As memórias não-voláteis (NVM ou *Non-Volatile Memory*) como o SSD (*Solid State Disk*), também têm evoluído, mas seu custo [Kasavajhala 2011], durabilidade e confiabilidade [Schroeder et al. 2016] continuam sendo impeditivos para a maioria das aplicações.

A vantagem em manter os dados na memória ao invés do disco está relacionada à latência de acesso a estes dados, pois remove-se a necessidade de acessar a camada mais lenta da hierarquia de memória, conforme demonstrado pela Figura 1 (adaptada de [Zhang et al. 2015]), que detalha as camadas de armazenamento, bem como uma estimativa de sua capacidade atual e da latência de acesso aos dados nela armazenados.

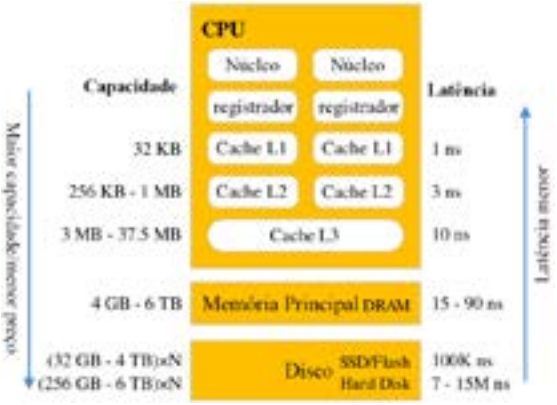

Figura 1. Hierarquia de memória.

Para que os dados possam ser processados pela CPU é necessário que estes estejam nos registradores e para tal é preciso que estes dados passem por todas as camadas da hierarquia de memória até chegarem aos registradores [Zhang et al. 2015]. Como pode ser visto na Figura 1, o disco é a camada mais distante e mais lenta, porém com a maior capacidade de armazenamento. Com o aumento da capacidade da camada de memória principal (composta pela memória RAM) os IMDB buscam trazer os dados para esta camada e evitar o nível mais lento da hierarquia de memória.

Dentre a miríade de bancos de dados com armazenamento em memória, os bancos chave-valor podem ser considerados os representantes mais versáteis, simples e com melhor desempenho, advindo principalmente da sua simplicidade [Pokorny 2013]. Portanto, muitos sistemas desta categoria sacrificam garantias de consistência em favor do desempenho [DeCandia et al. 2007] [Fowler 2015]. Nestes bancos, cada valor armazenado está vinculado a uma chave que identifica unicamente um valor [Han et al. 2011], sendo que este valor pode ser tanto um conteúdo binário quando uma estrutura de dados complexa, conforme as funcionalidades oferecidas pelo banco [Pokorny 2013]. Nas pró-

ximas seções são apresentados e discutidos os sistemas de armazenamento chave-valor em memória Redis, Memcached, Voldemort, Aerospike, Hazelcast e Riak KV.

3.1. Redis

Redis (*REmote DIctionary Server*) [Redis 2009] é um sistema de armazenamento de dados estruturados em memória que pode ser utilizado como banco de dados, *cache* e *message broker* [Cao et al. 2016]. Ele opera em um modelo cliente-servidor através de conexões TCP utilizando um protocolo próprio chamado RESP (REdis Serialization Protocol).

O modelo de dados do Redis é composto por cinco estruturas de dados diferentes para os valores (*string*, *list*, *set*, *sorted set* e *hash*), a persistência dos dados da memória em disco através de dois métodos (*snapshopts* chamados RDB e *append-only file* ou AOF) [Zhang et al. 2015]. A possibilidade de escalabilidade horizontal através do Redis Cluster foi adicionada apenas em 2015, na versão 3.0 do sistema. Segundo os autores de [Sanfilippo 2010], um sistema deve ser eficiente em um único nó quando for escalado.

3.2. Memcached

O Memcached [Memcached 2003] caracteriza-se como um sistema genérico de *cache* em memória. Ele foi construído pensando na melhoria de desempenho de aplicações *web* através da redução na demanda de requisições ao banco de dados em disco. Brad Fitzpatrick desenvolveu ele para melhorar o desempenho do site Livejournal.com através de uma solução melhor de *cache* [Galbraith 2009]. A sua implementação é na linguagem Perl e posteriormente reescrito em C. O Memcached utiliza uma arquitetura *multi-thread* e o controle de concorrência interno é feito através de uma *hash-table* estática de *locks* [Zhang et al. 2015].

A classificação do Memcached como banco de dados é discutível, uma vez que o mesmo não implementa persistência, failover [Galbraith 2009] nem escalabilidade horizontal, pois a distribuição dos dados entre múltiplas instâncias do sistema deve ser feita pelo cliente [Zhang et al. 2015]. O funcionamento do Memcached segue o modelo cliente-servidor e a comunicação ocorre através de conexões TCP ou UDP utilizando um protocolo próprio que suporta textos puros em ASCII ou dados binários [Soliman 2013].

3.3. Voldemort

O Voldemort [Voldemort 2009] foi desenvolvido pelo LinkedIn em linguagem Java com o objetivo de gerenciar funcionalidades dependentes de associações entre dados da rede social, tais como a recomendação de relacionamentos através da análise dos relacionamentos atuais [Sumbaly et al. 2012].

O Voldemort é inspirado no Dynamo, da Amazon [DeCandia et al. 2007], oferece comandos simples (put, get e delete) [Sumbaly et al. 2013] e uma arquitetura completamente distribuída, onde cada nó é independente e não existe um servidor principal de coordenação [Deka 2014]. O sistema é completamente modularizado e tanto a serialização dos dados quanto a persistência são oferecidas através de módulos plugáveis. Segundo [Sumbaly et al. 2012], a grande vantagem do Voldemort em relação ao Dynamo, é um mecanismo próprio de armazenamento desenhado para o pré-carregamento de grandes volumes de dados, em que o Voldemort passa a funcionar em modo somente leitura.

3.4. Aerospike

O Aerospike [Aerospike 2012] tem uma arquitetura modelada com foco em velocidade na análise de dados, escalabilidade e confiabilidade para aplicações *web*. Esse banco de

dados se apresenta como uma solução para a combinação de diferentes tipos de dados e também acessos por milhares de usuários. Pensando nisso, as suas operações são focadas em chave-valor e otimizadas para o uso da memória RAM em conjunto com memórias *flash* (NVM) [Aerospike 2012].

Diferente de vários de seus concorrentes, o próprio Aerospike disponibiliza bibliotecas de integração aos clientes, a fim de melhorar o desempenho na sua utilização. Quanto ao *cluster*, todos os nós são iguais, em uma arquitetura conhecida como *shared nothing*. A respeito do *server* é possível utilizar índices secundários e definir funções para otimizar a utilização dos dados. E por fim, a camada de armazenamento incorpora a utilização da memória RAM e de sistemas de armazenamento permanente.

3.5. Hazelcast

O Hazelcast [Hazelcast 2009] é uma ferramenta distribuída sob licença *open source* e comercial desenvolvida em Java. Possui seu foco em computação distribuída e escalabilidade horizontal, se destacando dos concorrentes por oferecer as garantias ACID (Atomicidade, Consistência, Isolamento e Durabilidade) dos bancos de dados relacionais tradicionais.

O *cluster* funciona em uma arquitetura *shared nothing*, onde não existe um ponto único de falha. Além de oferecer clientes para as linguagens comuns como Java, C, C++ e C#, o Hazelcast oferece uma API REST, está preparado para trabalhar com o protocolo de comunicação do Memcache e pode ser utilizado através do Hibernate [Hazelcast 2009].

3.6. Riak KV

O Riak KV [Basho 2009] possui como principal objetivo oferecer disponibilidade máxima, com escalabilidade horizontal em forma de *cluster*, sendo considerado um banco de dados de simples operação e fácil escalabilidade. Em sua versão comercial há suporte a *multi-cluster replication*, ou seja, é possível realizar a replicação de dados através de diferentes *clusters*, geograficamente distantes, a fim de reduzir a latência de acesso uniformemente para clientes através do globo.

Nota-se claramente a influência do Dynamo [DeCandia et al. 2007] no Riak KV, desde suas funcionalidades para execução distribuída até nas configurações do fator de replicação e arquitetura *shared nothing*.

4. Estudo Comparativo

Esta seção apresenta uma comparação entre os sistemas apresentados anteriormente. Para isso, as características foram classificadas em três grupos: (I) características mercadológicas, onde são explorados aspectos sem relação direta com funcionalidades ou o funcionamento do banco, tais como ano de lançamento, licenciamento e linguagem de desenvolvimento; (II) características do projeto, onde são descritas definições decididas no projeto do sistema, tais como escalabilidade, disponibilidade e consistência; e (III) características de manutenção, onde são explanadas os aspectos que tem relação direta com a manutenção e suporte ao sistema, tais como ferramentas internas para monitoramento e interface de gerenciamento.

Na Tabela 1 é possível avaliar: o ano em que a primeira versão do sistema foi lançada, os licenciamentos sob os quais o *software* é distribuído, a linguagem na qual o sistema foi desenvolvido, os sistemas operacionais suportados, as linguagens nas quais são oferecidos clientes para comunicação e os protocolos de comunicação suportados. A

partir dos dados disponibilizados, os interessados podem avaliar a maturidade do sistema, se a licença está alinhada com as necessidades empresariais e se a infraestrutura disponível e o esforço de implementação estão dentro do esperado.

Vale ressaltar que o Redis não suporta oficialmente Windows, mas uma versão para Windows x64 é mantida pela equipe da MS Open Tech (Microsoft Open Technologies). Quanto ao Voldemort e ao Hazelcast, como os mesmos rodam na JVM (Java Virtual Machine), podem-se considerar os sistemas operacionais suportados por esta tecnologia. Tanto Aerospike quanto Riak KV oferecem pacotes para sistemas baseados nas distribuições Linux Red Hat, Debian e Ubuntu. O Aerospike ainda oferece sua execução no OS X e Windows através de máquinas virtuais.

Tabela 1. Características mercadológicas

	Redis	Memcached	Voldemort	Aerospike	Hazelcast	Riak KV
Lançamento	2009	2003	2009	2012	2009	2009
Licenciamento	BSD-3 e comercial	BSD-3	Apache 2	AGPL e comercial	Apache 2 e comercial	Apache 2 e comercial
Desenvolvido	C	C	Java	C	Java	Erlang
SO Suporte	Linux, BSD, OSX e Windows	Debian/Ubuntu e Windows	JVM	Linux, OS X e Windows	JVM	Linux
Clientes	48 linguagens	Não existe listagem oficial	4 linguagens	12 linguagens	6 linguagens	21 linguagens
Protocolos	Próprio (RESP)	Próprio	HTTP, Socket, NIO	Próprio e JDBC	Próprio e Memcached	API HTTP e próprio

Quanto ao item linguagens com cliente, é importante notar que foram considerados apenas as linguagens e clientes listados no site oficial de cada sistema e que o site do Memcached não oferece uma listagem oficial das linguagens suportadas. Nota-se também que as linguagens C++, Java e Python são as únicas para as quais todos os sistemas possuem clientes. Os protocolos de comunicação são o meio de comunicação entre o cliente e o servidor, porém, na maioria dos casos a comunicação é feita através de um dos clientes já construídos e a empresa não precisa se preocupar com o protocolo utilizado pelo cliente.

Na Tabela 2 é possível avaliar: as opções para escalabilidade horizontal (ou *clusterização*), a classificação do sistema segundo o teorema CAP [Brewer 2000], como é feito o controle de concorrência, o suporte à transações ACID, as opções de persistência dos dados em disco, o suporte a dados complexos e as opções para autenticação do cliente. Analisando a Tabela 2 é possível avaliar se as funcionalidades e características do banco de dados atendem às demandas e características referentes aos dados que se pretende armazenar nos mesmos.

Quanto à escalabilidade, todos os bancos exceto o Memcached incluem suporte a *sharding* e replicação, sendo que a maioria (a exemplo do Dynamo [DeCandia et al. 2007]) oferecem fator de replicação configurável para leituras e escritas. O Redis utiliza o modelo master/slave, comumente utilizado nas bases de dados relacionais tradicionais.

O teorema CAP (*Consistency*, *Availability* e *Partition tolerance*) foi proposto por Eric Brewer em [Brewer 2000] e verificado em [Gilbert and Lynch 2002], desde então passou a ser largamente aceito pela academia. O teorema afirma que na existência de uma falha de comunicação (*partition*) cada nó de um sistema distribuído deve escolher entre responder requisições, mantendo a disponibilidade (*availability*) e assumindo o risco de não retornar os dados mais atuais, ou rejeitar requisições para garantir a consistência dos dados (*consistency*). Sistemas classificados como AP priorizam a disponibilidade, en-

quanto que sistemas classificados como CP priorizam a consistência. O teorema tem sido alvo de muitas críticas e Brewer explora algumas de suas limitações em [Brewer 2012], enquanto Abadi propõe o teorema PACELC como alternativa em [Abadi 2012].

Quanto à classificação do Redis, é importante mencionar que o ele não atende todos os requisitos de um sistema CP de [Brewer 2000], por usar replicação assíncrona entre os nós do *cluster*. O teorema CAP também não se aplica ao Memcached pelo fato de ele não suportar a criação de *clusters* e, portanto, não haver comunicação entre nós. O Riak KV possui uma configuração onde é possível definir qual dos atributos devem ser preservados (disponibilidade ou consistência).

Tabela 2. Características do projeto

	Redis	Memcached	Voldemort	Aerospike	Hazelcast	Riak KV
Escalabilidade horizontal	Mestre - Escravo	Não	Fator de Replicação	Fator de Replicação	Fator de Replicação	Fator de Replicação
Teorema CAP	CP	N/A	AP	AP	AP	Config.
Controle Concorrência	Single-thread	Mutex lock	MVCC	Test-and-set	Multi-single-thread	MVCC
Transações	Parcial	Não	Não	Parcial	Sim	Não
Persistência em disco	RDB e AOF	Não	Config.	Assínc.	Banco auxiliar	Banco auxiliar
Suporte a dados complexos	Sim	Não	Sim	Sim	Sim	Sim
Autenticação	Simples	SASL	Kerberos	Somente comercial	Simples, SSL, Kerberos,IP	Sim, e autorização

O controle de concorrência é implementado de diferentes maneiras. No Redis a execução é *single-thread* e as requisições são processadas de forma assíncrona internamente [Zhang et al. 2015], sendo que apenas um *master* responde por uma determinada chave, portanto não há concorrência. O Memcached utiliza *mutex* (*mutual exclusive*) *lock*. Tanto Voldemort quanto Riak KV seguem a implementação do Dynamo [DeCandia et al. 2007] e utilizam *vector clocks*, uma implementação do versionamento baseado em locking otimista conhecida por MVCC (Multi Version Concurrency Control). O Aerospike utiliza o método conhecido como *test-and-set* ou *check-and-set* (CAS), uma operação atômica implementada a baixo nível que escreve em um local de memória e retorna o valor antigo. No Hazelcast, é criada uma *thread* para atender cada uma das partições internas de dados, portanto ainda que ele seja *multi-thread*, uma chave específica está num contexto *single-thread* e, portanto, não há concorrência.

Quanto as transações, há um nível bastante variado de suporte oferecido pelos sistemas. Ainda que o Redis tenha suporte básico a transações, estas não possuem opção de rollback e a durabilidade da mesma depende da persistência em disco. Memcached, Voldemort e Riak KV declaram não suportarem transações, enquanto que o Aerospike suporta transações que envolvam uma única chave ou que sejam somente leitura, no caso de envolverem múltiplas chaves. O Hazelcast se destaca sendo o único a oferecer transações ACID completas.

A persistência em disco é oferecida por todos os bancos, exceto o memcached. No Redis são oferecidas duas formas complementares: *snapshot* (RDB), onde todos os dados na memória são gravados em disco, e *append-only file* (AOF), onde cada operação é gravada em um arquivo de log e o arquivo é reescrito quando chega em um tamanho pré-determinado. O Voldemort oferece opções para configurar a persistência como síncrona (*write through*, onde a operação é persistida antes do retorno ao cliente) ou assíncrona (*write behind*, onde o cliente recebe a confirmação e posteriormente a operação é persistida), enquanto que o Aerospike faz a persistência de forma assíncrona. Vale mencionar

que o Aerospike tem melhorias focadas no uso de SSD como dispositivo de armazenamento permanente. Tanto Hazelcast como Riak KV oferecem persistência através do acoplamento de um banco de dados auxiliar e a assincronicidade é configurável.

Referente ao suporte a dados complexos, vale notar que todos os sistemas, exceto o Memcached, suportam chaves do tipo lista e hashtable (ainda que com nomes diferentes). Hazelcast e Voldemort se baseiam fortemente nas classes do Java, Redis e Riak KV ainda oferecem suporte ao tipo HyperLogLogs, enquanto Redis e Aerospike oferecem suporte a tipagem ou comandos relativos a georreferenciamento.

Finalizando, enquanto que Redis oferece autenticação simples através de credenciais pré-configuradas, o Memcached oferece autenticação através do protocolo SASL (*Simple Authentication and Security Layer*). O Voldemort é integrado ao protocolo Kerberos. O Aerospike oferece autenticação apenas em sua versão com licenciamento comercial. O Hazelcast oferece todos os protocolos supracitados e o SSL. Por último, o Riak KV oferece um sistema próprio de usuários e grupos, com autenticação e autorização baseada em diversos mecanismos, incluindo senhas e certificados digitais.

Na Tabela 3 está descrita a existência de interfaces de gerenciamento, ferramentas de monitoramento e benchmarks para os sistemas estudados. É importante notar que foram avaliadas apenas as ferramentas oficiais dos desenvolvedores dos sistemas. Portanto, muitas destas ferramentas, ainda que não estejam declaradas aqui, já foram desenvolvidas pela comunidade e estão disponíveis. Estas informações são particularmente interessantes para avaliar o esforço de manutenção que será despendido após a implantação do sistema.

Tabela 3. Características de manutenção

	Redis	Memcached	Voldemort	Aerospike	Hazelcast	Riak KV
Interface de Gerenciamento	Não	Não	Básica	À parte	Comercial	Sim
Ferramentas de Monitoramento	'INFO'	'stats'	JMX	'asadm'	JMX	'stats'
Benchmark embutido	Sim	Não	Sim	Sim	Não	Sim

Quanto às interfaces de gerenciamento oferecidas pelos sistemas avaliados, o Redis e o Memcached são os únicos que não as oferecem nativamente (ainda que existam opções na comunidade) e o Voldemort oferece uma interface básica à parte escrita em Ruby, que está sem manutenção. O Aerospike oferece uma interface que deve ser instalada à parte. No Hazelcast, esta funcionalidade está disponível apenas na versão comercial. O Riak KV é o único onde a interface de gerenciamento já está integrada ao código principal do programa, não requerendo nenhuma instalação extra.

A respeito de ferramentas de monitoramento, o Redis oferece comandos como *INFO*, *MEMORY* e *LATENCY*, enquanto que o Memcached oferece o comando *stats* e o Aerospike oferece o comando *asadm*. O Voldemort possui uma interface completa de monitoramento exposta através de *Java Management Extensions* (JMX). Esta é a mesma estratégia utilizada pelo Hazelcast. O Riak KV oferece os comandos *stat* e *stats* em sua interface de linha de comando (CLI) *riak-admin* e a URL */stats* em sua API HTTP.

No item *benchmark* embutido foi avaliado se são disponibilizados *benchmarks* junto com o sistema, cujo o principal objetivo é avaliar o desempenho do sistema em determinada infraestrutura. Enquanto que Memcached e Hazelcast não oferecem ferramentas próprias para realização de *benchmark*, no Redis existe a ferramenta *redis-benchmark* e o Voldemort oferece a *voldemort-performance-tool*. No Aerospike os *benchmarks* estão nos clientes disponibilizados e no Riak KV o nome dado ao *benchmark* é *Basho Bench*.

Após comparar as características dos bancos de dados chave-valor com armazenamento em memória, Redis, Memcached, Voldemort, Aerospike, Hazelcast e Riak KV, é fácil entender o motivo pelo qual o Memcached não é considerado um banco de dados, pois seu foco destoa bastante de seus semelhantes. É possível perceber também que, mesmo que o Redis tenha oferecido suporte à clusterização em suas versões mais recentes, os outros sistemas ainda estão à frente quando o assunto é funcionalidades para execução distribuída. É possível perceber também como Voldemort e Hazelcast se utilizam do ecossistema Java para prover funcionalidades interessantes e como o paper do Dynamo [DeCandia et al. 2007] influencia principalmente Voldemort e Riak KV.

5. Conclusões

Após estudar os bancos chave-valor com armazenamento em memória, é possível notar que mesmo um subconjunto específico de bancos NoSQL traz muitas variáveis. Neste sentido, destaca-se a grande quantidade de características que devem ser cuidadosamente avaliadas pelo analista para a correta tomada de decisão quanto ao banco mais adequado às necessidades. Dentre estas características, destacam-se o padrão de busca e gravação de dados da aplicação cliente, a importância da durabilidade dos dados, o comportamento desejado frente a partições no *cluster*, o ambiente de infraestrutura onde a solução será implantada, o ambiente de desenvolvimento da aplicação cliente e as perspectivas de crescimento na demanda da aplicação.

Com as características dos sistemas esclarecidas, percebe-se que outro fator importante para a escolha do banco de dados chave-valor com armazenamento em memória a ser adotado é o desempenho, que é justamente o ponto que traz mais interesse a esta categoria de sistemas. Como trabalho futuro, propõe-se uma avaliação do desempenho dos sistemas aqui estudados, comparando o desempenho das variadas características compartilhadas pelos mesmos.

Referências

[Abadi 2012] Abadi, D. (2012). Consistency Tradeoffs in Modern Distributed Database System Design: CAP is Only Part of the Story. *Computer*, 45(2):37–42.

[Aerospike 2012] Aerospike (2012). Aerospike | High Performance NoSQL Database. Access on <http://www.aerospike.com/>.

[Basho 2009] Basho (2009). Riak KV. Access on <http://basho.com/products/riak-kv/>.

[Brewer 2000] Brewer, E. (2000). Towards Robust Distributed Systems. In *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '00, pages 7–, New York, NY, USA. ACM.

[Brewer 2012] Brewer, E. (2012). CAP twelve years later: How the "rules"have changed. *Computer*, 45(2):23–29.

[Cao et al. 2016] Cao, W., Sahin, S., Liu, L., and Bao, X. (2016). Evaluation and Analysis of In-Memory Key-Value Systems. In *2016 IEEE International Congress on Big Data (BigData Congress)*, pages 26–33.

[Carlson 2013] Carlson, J. L. (2013). *Redis in Action*. Manning, Shelter Island, NY, USA.

[Chang et al. 2008] Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., and Gruber, R. E. (2008). Bigtable: A distributed storage system for structured data. *ACM Trans. on Computer Systems (TOCS)*, 26(2):4.

[DeCandia et al. 2007] DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., and Vogels, W. (2007). Dynamo: amazon's highly available key-value store. *ACM SIGOPS operating systems review*, 41(6):205–220.

[Deka 2014] Deka, G. C. (2014). A survey of cloud database systems. *IT Professional*, 16(2):50–57.

[Fowler 2015] Fowler, A. (2015). *NoSQL For Dummies*. John Wiley & Sons, 111 River Street, Hoboken, New Jersey, USA.

[Galbraith 2009] Galbraith, P. (2009). *Developing Web Applications with Apache, MySQL, memcached, and Perl*. John Wiley & Sons.

[Gilbert and Lynch 2002] Gilbert, S. and Lynch, N. (2002). Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-tolerant Web Services. *SIGACT News*, 33(2):51–59.

[Han et al. 2011] Han, J., E, H., Le, G., and Du, J. (2011). Survey on NoSQL database. In *2011 6th International Conference on Pervasive Computing and Applications*, pages 363–366.

[Hazelcast 2009] Hazelcast (2009). Hazelcast the Leading In-Memory Data Grid - Hazelcast.com. Access on <https://hazelcast.com/>.

[Hecht and Jablonski 2011] Hecht, R. and Jablonski, S. (2011). NoSQL evaluation: A use case oriented survey. In *2011 International Conference on Cloud and Service Computing (CSC)*, pages 336–341.

[Kasavajhala 2011] Kasavajhala, V. (2011). Solid State Drive vs. Hard Disk Drive Price and Performance Study. *Proc. Dell Technical White Paper*, pages 8–9.

[Memcached 2003] Memcached (2003). memcached - a distributed memory object caching system. Access on <https://memcached.org/>.

[Pokorny 2013] Pokorny, J. (2013). NoSQL databases: a step to database scalability in web environment. *International Journal of Web Information Systems*, 9(1):69–82.

[Redis 2009] Redis (2009). Redis.io. Access on <http://redis.io/>.

[Robbins 2008] Robbins, S. (2008). RAM is the new disk... Access on <https://www.infoq.com/news/2008/06/ram-is-disk>.

[Sanfilippo 2010] Sanfilippo, S. (2010). On Redis, Memcached, Speed, Benchmarks and The Toilet . Access on <http://antirez.com/post/redis-memcached-benchmark.html>.

[Schroeder et al. 2016] Schroeder, B., Lagisetty, R., and Merchant, A. (2016). Flash Reliability in Production: The Expected and the Unexpected. In *Proceedings of the 14th USENIX Conference on File and Storage Technologies (FAST '16)*, FAST '16, pages 67–80, Santa Clara, CA, USA.

[Soliman 2013] Soliman, A. (2013). *Getting Started with Memcached*. Packt.

[Sumbaly et al. 2012] Sumbaly, R., Kreps, J., Gao, L., Feinberg, A., Soman, C., and Shah, S. (2012). Serving large-scale batch computed data with Project Voldemort. In *Proceedings of the 10th USENIX conference on File and Storage Technologies*, page 18.

[Sumbaly et al. 2013] Sumbaly, R., Kreps, J., and Shah, S. (2013). The Big Data Ecosystem at LinkedIn. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 1125–1134, New York, NY, USA. ACM.

[Voldemort 2009] Voldemort (2009). Project Voldemort. Access on <http://www.project-voldemort.com/>.

[Zhang et al. 2015] Zhang, H., Chen, G., Ooi, B. C., Tan, K.-L., and Zhang, M. (2015). In-Memory Big Data Management and Processing: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 27(7):1920–1948.

Artigos de Aplicações e Experiências

MahoutGUI: Uma Interface Gráfica para Gerar Recomendações com o Apache Mahout Diretamente de Banco de Dados usando Mapeamento Objeto-Relacional

Gláucio R. Vivian[1], **Cristiano R. Cervi**[1]

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

{149293,cervi}@upf.br

Abstract. *The Recommender Systems are responsible by assisting the users in information filtering process. This work reports the development of a graphical user interface to recommender of Apache Mahout. We also propose a java class to access the data stored in relational data base through Object-Relational Mapping using the Hibernate framework. The first contribution of this work is the simplification of the user interaction with system, obtained through the graphical user interface. The second is the flexibility and portability to different relational data base managers systems obtained through Hibernate.*

Resumo. *Os Sistemas de Recomendações são responsáveis por auxiliar os usuários no processo de filtragem de informações. Este trabalho relata o desenvolvimento de uma interface gráfica para o recomendador do Apache Mahout. Também propomos uma classe java de acesso aos dados armazenados em banco de dados relacionais por meio do mapeamento objeto-relacional usando o framework Hibernate. A primeira contribuição deste trabalho é a simplificação da interação do usuário com o sistema, obtida pela interface gráfica. A segunda é a flexibilidade e portabilidade para diferentes gerenciadores de banco de dados obtida por meio do Hibernate.*

1. Introdução

Os Sistemas de Recomendação estão amplamente difundidos, seja no comércio eletrônico, redes sociais, notícias, conteúdo sob demanda e muitas outras aplicações. Um dos mais importantes projetos de Sistema de Recomendação é o Apache Mahout[1]. Segundo [Owen et al. 2012], trata-se de um projeto *open source* com recursos de aprendizado de máquina e mineração de dados para ambientes de execução distribuída baseados no Apache Hadoop[2]. A paralelização é obtida por meio do paradigma de programação MapReduce apresentado por [Dean e Ghemawat 2008]. Os dados são armazenados em um sistema de arquivos distribuídos com tolerância a falhas chamado de *Hadoop Distributed File System* (HDFS).

O projeto conta com diversas e modernas técnicas de Classificação, Recomendação e Agrupamento (Clustering). As técnicas de recomendação disponibilizadas são fatoração de matriz (SVD) e filtragem colaborativa usuário-item e item-item.

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

A leitura dos dados é abstraída por meio da classe DataModel, que pode buscá-los diretamente em arquivos no formato CSV ou por meio de conexões JDBC. Os parâmetros de configuração são bastante flexíveis, permitindo utilizar diversas técnicas de similaridades. No caso da técnica usuário-item, podem localizar os usuários por meio do algoritmo do vizinho mais próximo (*Nearest Neighbor*) baseado em limiar (*threshold*) ou n quantidades. Também existem algumas métricas (*Precision, Recall*, RMSE e MAE) para avaliar os resultados, permitindo dessa forma a realização completa de experimentos avaliativos. Uma deficiência no projeto é a inexistência de interface gráfica, sendo tudo realizado por meio do *shell* do Sistema Operacional ou biblioteca Java.

O *framework* Hibernate[3] possibilita um alto nível de abstração no acesso a informações armazenadas em banco de dados relacionais por meio da técnica de mapeamento objeto-relacional. Esta se caracteriza por representar os dados na forma de objetos, atributos e coleções do paradigma de programação orientada a objetos. Para o desenvolvedor, as particularidades do SGBD são todas abstraídas. Existe uma linguagem próxima da tradicional SQL, denominada de HQL que possibilita a execução de consultas.

O objetivo deste trabalho é apresentar uma interface gráfica de usuário que gera recomendações com o Apache Mahout. Além disso, construiu-se uma classe java para abstrair o acesso à diferentes SGBDs por meio do mapeamento objeto-relacional com o *framework* Hibernate.

Este artigo está organizado da seguinte forma: Na seção 2 são analisados alguns trabalhos correlatos. Na seção 3 é apresentado o aplicativo. Na seção 4 são apresentados os experimentos e resultados. Finalmente, na seção 5, são apresentadas as conclusões e trabalhos futuros.

2. Trabalhos Correlatos

[Akbarnejad et al. 2010] apresentam o QueRIE, trata-se de um sistema de recomendação que possibilita aos usuários gerarem consultas personalizadas em linguagem SQL para grandes bancos de dados relacionais. O sistema é constituído por dois diferentes motores de execução. O primeiro, chamado *Tuple-based*, identifica partes com interesse em potencial da base de dados que foram acessados por usuários similares no passado. O segundo, denominado *Fragment-based*, identifica consultas similares que outros usuários postaram para o usuário atual. Os experimentos foram conduzidos em dois diferentes cenários de consultas SQL: simples e complexas. Como resultados, demonstrou-se que a proposta habilita os usuários a gerarem recomendações com o SGBD SkyServer por meio da análise dos seus logs de consultas executadas.

Nos trabalhos de [Sarwat 2012, Sarwat et al. 2013] é apresentado o RecDB. Trata-se de um *fork* do Postgresql com a adição de um *engine* baseado no LensKit[4] do GroupLens[5] com os recursos para gerar recomendações usando a técnica de filtragem colaborativa do tipo item-item e usuário-item. O *engine* é constituído pelos seguintes módulos: i) Rec-Store: módulo responsável pelo armazenamento, manutenção e otimização dos dados. ii) Rec-tree: eficiente estrutura em árvore responsável e pela indexação dos atributos

[3]http://hibernate.org/orm/
[4]http://lenskit.grouplens.org/
[5]http://grouplens.org/

dos usuários/itens. iii) Rec-Query: processador de consultas SQL. Além disso, foi apresentada a adição de recursos ao padrão SQL com o objetivo de definir recomendadores e executar consultas com a solicitação de recomendações. Os autores colocam que a sua proposta apresenta as seguintes características: usabilidade, flexibilidade, facilidade de integração e eficiência. Foram realizados dois exemplos demonstrativos de aplicação, o primeiro com um sistema de recomendações para restaurante e o segundo sobre filmes.

3. A Aplicação Proposta

A aplicação proposta denominada MahoutGUI[6], deve simplificar a interação do usuário com as diversas opções apresentadas de recomendação. Na Figura 1 pode-se visualizar a interface gráfica da aplicação proposta.

(a) (b) (c)

Figura 1. Interface Gráfica da aplicação MahoutGUI.

Em (a) apresenta-se a entrada dos dados, em (b) as configurações e finalmente em (c) as recomendações geradas.

Além da interface gráfica proposta, desenvolveu-se uma classe de acesso a banco de dados utilizando-se o mapeamento objeto-relacional com o *framework* Hibernate. Isto possibilita que a aplicação seja mais flexível entre diversos gerenciadores de banco de dados. Na Figura 2 pode-se visualizar o diagrama de classes da aplicação proposta.

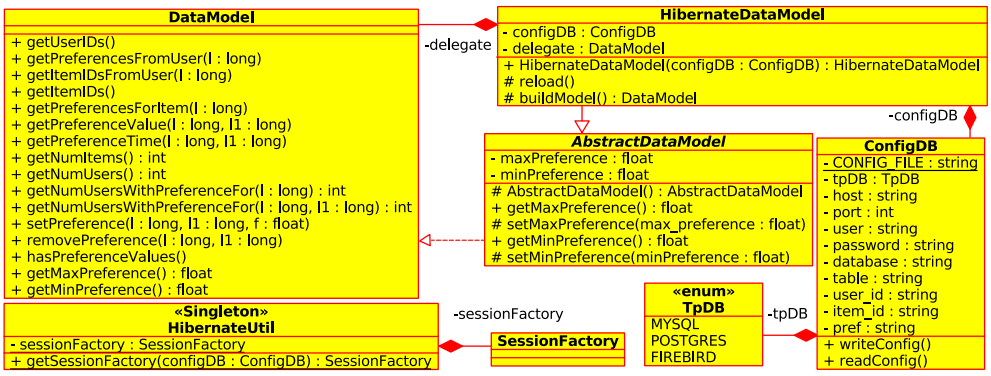

Figura 2. Diagrama de Classes para o mapeamento objeto-relacional.

A classe HibernateDataModel é responsável pelo mapeamento objeto-relacional da aplicação. A mesma herda de AbstractDataModel, que por sua vez implementa DataModel. A classe ConfigDB é responsável por armazenar as informações de acesso ao

[6]https://github.com/grvivian/MahoutGUI

banco de dados. A enumeração TpDB define os SGBDs suportados pela aplicação proposta. A classe HibernateUtil implementa o *design pattern* Singleton, responsável por unificar o acesso ao atributo sessionFactory do Hibernate.

4. Experimentos e Resultados

Para testar a aplicação desenvolvida, utilizou-se um banco de dados PostgreSQL 9.5 com dados do projeto MovieLens de [Harper e Konstan 2016]. Ele contém 100.004 avaliações realizadas por 671 usuários sobre 9.125 filmes. Foram realizados dois testes solicitando 5 recomendações para os usuários 460 e 246 (escolhido aleatoriamente). Na Figura 3 pode-se visualizar as recomendações geradas.

(a) (b)

Figura 3. Resultado obtidos.

Em (a) foi utilizada filtragem colaborativa item-item usando a similaridade Log-Likelihood. Em (b) utilizou-se filtragem colaborativa usuário-item, com a mesma similaridade e com os 10 vizinhos mais próximos.

5. Considerações Finais e Trabalhos Futuros

Este trabalho apresentou o desenvolvimento do MahoutGUI, trata-se de uma interface gráfica para o Apache Mahout. A mesma possibilita de forma visual e intuitiva a realização rápida de experimentos com as técnicas de filtragem colaborativa. Além disso, desenvolveu-se uma classe para acesso à informações diretamente em banco de dados relacionais por meio do mapeamento objeto-relacional. Dessa forma, a aplicação proposta simplifica a realização de experimentos com sistemas de recomendação, possibilitando o acesso a usuários não especialistas da área. Como sugestão de trabalhos futuros, pretende-se suportar técnicas de fatoração de matriz e recomendações para usuários anônimos.

Referências

Akbarnejad, J., Chatzopoulou, G., Eirinaki, M., Koshy, S., Mittal, S., On, D., Polyzotis, N., e Varman, J. S. V. (2010). Sql querie recommendations. *Proceedings of the VLDB Endowment*, 3(1-2):1597–1600.

Dean, J. e Ghemawat, S. (2008). Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.

Harper, F. M. e Konstan, J. A. (2016). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19.

Owen, S., Anil, R., Dunning, T., e Friedman, E. (2012). *Mahout in action*. Manning Shelter Island, NY.

Sarwat, M. (2012). Recdb: towards dbms support for online recommender systems. In *Proceedings of the on SIGMOD/PODS 2012 PhD Symposium*, páginas 33–38. ACM.

Sarwat, M., Avery, J., e Mokbel, M. F. (2013). Recdb in action: recommendation made easy in relational databases. *Proceedings of the VLDB Endowment*, 6(12):1242–1245.

Desenvolvimento do Sistema de Acompanhamento do Fluxo de Demandas do Plano de Ação do Instituto Federal Farroupilha Campus Alegrete

Lenon Ricardo Machado de Souza[1], Marta Breunig Loose[2]

[1]Instituto Federal Farroupilha -- Campus Alegrete – (IFFar Campus Alegrete)
97.555-00 – Passo Novo – RS – Brasil

[2]Instituto Federal Farroupilha -- Campus Santo Ângelo – (IFFar Campus Santo Ângelo)
98.806-700 – Indúbras – RS – Brasil

lenonrmsouza@gmail.com, marta.breunig@iffarroupilha.edu.br

Abstract. *The Action Plan at the Federal Institute Farroupilha Campus Alegrete consists in a sectors demands list. The demands are goods and services purchase requests, which have a complex processing flow and are not always properly monitored. This paper describes the web system development that will allow the goods requests inclusion and also the demands flow monitoring of the Action Plan. The system uses the graph model, which is a NoSQL database category. The main reason for choosing this model is the characterization of the system stored data, being basically generated documents among several flow steps.*

Resumo. *O Plano de Ação no Instituto Federal Farroupilha Campus Alegrete consiste em uma lista de demandas de um setor. As demandas são pedidos de aquisição de bens e serviços, que possuem um fluxo de tramitação complexo e nem sempre tem o acompanhamento adequado. Este trabalho descreve o desenvolvimento de um sistema web que permitirá a inclusão de solicitações de bens e também o acompanhamento do fluxo das demandas referentes ao Plano de Ação. O sistema utiliza o modelo de grafos, que é uma categoria de banco de dados NoSQL. O principal motivo da escolha desse modelo é a caracterização dos dados armazenados no sistema, sendo basicamente documentos gerados ao longo de fluxo de várias etapas.*

1. Introdução

Anualmente no Instituto Federal Farroupilha Campus Alegrete (IFFar Campus Alegrete) é realizado o planejamento estratégico, que consiste na definição de ações de cada direção junto as suas coordenações através de diversas reuniões de viabilidade. Nessas reuniões são especificadas as prioridades que farão parte do Plano de Ação (PA) do próximo ano.

Atualmente, a solicitação das demandas é realizada manualmente, o que origina alguns problemas ao longo das interações com os setores envolvidos na verificação, validação e execução. Basicamente, esses problemas envolvem a dificuldade do solicitante na localização do documento, problemas de comunicação e falta de informações quanto ao atendimento das demandas.

Observando tais problemas, este trabalho descreve o desenvolvimento de um software que auxilia o acompanhamento da execução das demandas propostas no PA. Nesse sistema, os usuários poderão incluir solicitações de bens, além de visualizar a situação atual, o fluxo e os detalhes do atendimento das demandas. O sistema proporcionará o registro digital das demandas, permitindo também a definição de prazos para o atendimento em cada etapa, o que ajudará a diminuir os problemas citados anteriormente.

2. Banco de Dados Não-Relacional

Aproximadamente no ano de 2009 surgiu o conceito NoSQL, que busca suprir as necessidades do modelo relacional, contando com a alta performance e a rápida replicação de dados. As principais características desse modelo são a escalabilidade e a velocidade nas buscas dos valores armazenados.

Atualmente existem vários modelos de banco de dados não-relacionais disponíveis, sendo que cada um possui conceitos e características próprias, possibilitando sua aplicação de acordo com a necessidade dos desenvolvedores. Dentre os tipos de bancos de dados NoSQL estão os modelos chave-valor, documentos, colunas e grafos.

Especificamente, o modelo de grafos promove diversas soluções inovadoras para o armazenamento e processamento de grandes quantidades de dados. Segundo Vieira et al. (2012), tais soluções foram propostas devido a alguns problemas gerados por aplicações na Web 2.0, às quais necessitavam operar com grande volume de dados e uma arquitetura flexível, capaz de trabalhar com vários nós de processamento sem a necessidade de adicionar mais máquinas físicas.

Em relação à consistência desse modelo de banco de dados, nodos só poderão ser excluídos caso não exista nenhuma relação ligada a ele, garantindo que dados importantes não sejam perdidos acidentalmente. Não existem restrições em relação aos valores que podem ser armazenados no grafo, o que possibilita a criação de um nó com qualquer valor ou propriedade. Segundo Sadalage e Fowler (2013), um dos recursos interessantes dos bancos de dados de grafos é encontrar caminhos entre dois nós e dessa forma determinar suas características.

No sistema descrito nesse trabalho é utilizado o modelo de grafos. Sua escolha é mais adequada pois o fluxo do sistema possui características que remetem a esse modelo, como as etapas e seus respectivos documentos, que podem ser representadas pelos nós de um grafo. Já as ações de cada etapa possuem características semelhantes às arestas de um grafo. Neste trabalho é utilizada a ferramenta Neo4J para a criação dos grafos, em conjunto com a linguagem PHP além de outras tecnologias voltadas para o desenvolvimento web.

3. Estudo de Caso

No IFFar Campus Alegrete, o Plano de Ação consiste em um documento onde são registradas todas as necessidades anuais de um setor ou coordenação, sendo elaborado pelos coordenadores correspondentes a cada eixo tecnológico. O documento é enviado para os setores responsáveis pela avaliação, momento em que as demandas são estudadas a fim de verificar sua viabilidade e desta maneira as aquisições realizam-se ao longo do ano.

Porém todo esse processo é realizado de forma física e manual, ou seja, é necessário se trabalhar com documentos em papel e todo fluxo deve ser realizado e atendido pelas pessoas envolvidas, causando assim certos problemas. A dificuldade de comunicação entre as partes envolvidas, principalmente em relação ao atendimento as demandas, é o maior deles. Dessa forma, o sistema descrito neste trabalho auxiliará no gerenciamento do fluxo de execução do Plano de Ação digitalmente. Assim, o setor responsável por alguma etapa do atendimento das demandas participará do fluxo executando ações, tais como receber, encaminhar ou retornar, além de documentar e acompanhar a situação das mesmas.

4. Resultados e Discussões

Através do levantamento de requisitos foi possível especificar um fluxo para ser aplicado ao software. Conforme mostra a Figura 1, a demanda é iniciada a partir do pedido do usuário e posteriormente são realizadas outras etapas até a finalização daquela solicitação.

Figura 1. Fluxo resumido do sistema

O fluxo apresentado na Figura 1 foi resumido em sete etapas, para possibilitar a identificação do funcionamento do sistema proposto, porém no fluxo completo são realizadas o total de vinte e cinco etapas. Sendo que a duração aproximada de uma demanda no sistema é de um ano e seis meses.

Através da Figura 2 é possível visualizar a solicitação de bens, sendo que o solicitante preenche as informações a partir dos dados das demandas presentes no Plano de Ação do seu setor. Com isso, o documento da solicitação segue para a etapa de Avaliações (conforme Figura 1). Após a verificação da sua viabilidade, um novo documento é gerado a partir da solicitação original, com dados complementares e alterações necessárias para o prosseguimento do fluxo.

Figura 2. Tela de Solicitação da Demanda

5. Conclusões

O modelo de banco de dados de grafos, comparado ao modelo relacional, possui alguns benefícios que são importantes para a elaboração do projeto apresentado. Esses benefícios consistem na escalabilidade, velocidade nas consultas e tolerância a falhas. Tais características são essenciais para o desenvolvimento, além de possibilitar a manipulação de grandes quantidades de dados sem perder o desempenho.

A fase atual do projeto está compreendida no desenvolvimento da integração da linguagem PHP com a ferramenta Neo4J. Com o sistema será possível melhorar a execução das demandas, pois as etapas são realizadas digitalmente, facilitando a geração de documentos, a definição de status e indicação de observações pelos usuários. Uma vantagem importante será a agilidade na busca por informações de determinada demanda, já que através do sistema os usuários poderão verificar essas informações de maneira simplificada. Por fim, é importante ressaltar que a utilização do modelo de grafos para armazenar dados do sistema contribui para o desenvolvimento e a difusão dos bancos de dados NoSQL.

Referências

Lima, Claudio de; Mello, Ronaldo S. "Um Estudo sobre Modelagem Lógica para Bancos de Dados NoSQL."

Vieira, Marcos Rodrigues; et al. "Bancos de Dados NoSQL: conceitos, ferramentas, linguagens e estudos de casos no contexto de Big Data." *Simpósio Brasileiro de Bancos de Dados* (2012).

Sadalage, Pramod J.; Fowler, Martin. NoSQL Essencial: Um guia conciso para o Mundo emergente da persistência poliglota. Novatec Editora, 2013.

Consulta de Dados Espaciais em um Sistema de Informações de uma Bacia Hidrográfica

Vania Elisabete Schneider[1], Odacir Deonisio Graciolli[2], Helena Graziottin Ribeiro[3], Roberto Canuto Spiandorello[4], Guilherme Vanzin Hoffmann[5], Miguel Ângelo Pontalti Giordani[6]

[1,4,5,6]Instituto de Saneamento Ambiental – Universidade de Caxias do Sul (UCS)
CEP 95070-560 – Caxias do Sul – RS – Brazil

[2,3]Área de conhecimento de Ciências Exatas e Engenharias – Universidade de Caxias do Sul (UCS)
CEP 95070-560 – Caxias do Sul – RS – Brazil

veschnei@ucs.br, odgracio@ucs.br, hgrib@ucs.br, rcspiandorello@ucs.br, gvhoffma@ucs.br, mapgiordani@ucs.br

Abstract. *This article presents the gains obtained in an Information System of a river basin with the application of an extension association for spatial data manipulation in a database management system. It presents some spatial data manipulation processes and their applications, which will ultimately allow a better management decision-making, regarding the monitoring of phenomena inside the limits of the Taquari-Antas basin.*

Resumo. *Este artigo apresenta os ganhos obtidos em um Sistema de Informações de uma bacia hidrográfica com a associação da extensão para manipulação de dados espaciais PostGIS em um sistema gerenciador de banco de dados. Apresenta alguns processos de manipulação de dados espaciais e suas aplicações, que irão em última instância, permitir uma melhor tomada de decisões gerenciais, no que se refere ao acompanhamento dos fenômenos ocorridos dentro dos limites da bacia Taquari-Antas.*

1. Introdução

O Sistema de Informações em questão é uma demanda das pequenas centrais hidrelétricas (PCH) de uma bacia hidrográfica. Esse sistema foi desenvolvido para armazenar dados gerados em monitoramentos de qualidade da água, climatologia e fauna das diferentes instituições instaladas na bacia, visando assegurar a análise semântica e espacial, e também tem a função de gerar relatórios de acordo com as necessidades da gestão ambiental.

Os dados gerados nos monitoramentos de cada sub-bacia são armazenados no PostgreSQL - sistema de gerenciamento de banco de dados relacional, que apresenta recursos orientados a objetos como extensão do modelo relacional e fornece escalabilidade e suporte para tipos de dados complexos (objetos grandes, dados multimídia, dados espaciais, etc.) combinando os modelos relacional e orientado a objetos [1].

O PostgreSQL é habilitado para manipulação de dados espaciais através do PostGIS - um extensor de banco de dados espaciais, compatível com a OGC (*Open Geo Consortium*). É uma ferramenta livre e de código aberto que, segundo [Singh, S. P. & P. Singh 2014], contribui para uma implementação rápida e com pouco ou nenhum custo de software. O PostGIS adiciona funções espaciais para análise de componentes geométricos, manipulação de geometrias e determinação de relações espaciais.

O sistema utiliza o PostgreSQL com o PostGIS, mas algumas consultas exigiram um tratamento específico para manipulação de dados geográficos que não eram contemplados anteriormente. Essas consultas anteriores não estabeleciam relações entre os diferentes objetos espaciais (pontos, linhas, polígonos), resultando na subutilização das colunas para armazenamento de coordenadas geográficas. Este artigo apresenta o processo de inclusão de novas consultas ao sistema de informações para manipulação de dados espaciais, para atender a consultas que precisavam obter a localização de determinados pontos de monitoramento de dados relacionando-os a sua sub-bacia.

2. Banco de dados espacial no contexto do sistema de informação em questão

Os dados espaciais utilizados pelo sistema são pontos, linhas e polígonos, utilizados para representar visualmente os limites da bacia, das sub-bacias, pontos de fauna, pontos de qualidade da água, e pontos de barramento. Eles são armazenados e formatados conforme o código padrão internacional ou *Spatial Reference System Identifier* (SRID)[5].

O PostGIS - banco de dados espaciais, contém operadores que manipulam os objetos geométricos mais diretamente. Esse banco armazena e manipula objetos espaciais como qualquer outro objeto no banco de dados. Pode-se verificar se um ponto intercepta uma linha, se duas linhas se cruzam, se uma linha está completamente contida em um polígono, se um polígono está contido em outro, o que e é bem complicado de fazer se na consulta essas operações são escritas apenas usando pontos: a consulta pode ficar ineficiente.

O PostGIS permite solucionar problemas que abrangem análises baseadas no posicionamento dos objetos (pontos, linhas, polígonos) inseridos nas sub-bacias, estendendo as relações até então obtidas somente através de relacionamentos explícitos, como com a tabela empreendimento. Aplicações para a área ambiental incluem a aplicação de funções de interseção e distância para acompanhamento de áreas desflorestadas e cálculos de distâncias entre corpos de água [3]. Este extensor provê um "índice espacial", capaz de identificar objetos contidos dentro de outro e que estão dispostos em um espaço bidimensional, diferentemente dos índices tradicionais cuja ordenação é feita por strings, números ou datas [4]. O índice espacial, portanto, é utilizado para determinar a relação entre as geometrias selecionadas de maneira rápida: a relação é feita indexando a caixa delimitadora da geometria em vez da própria geometria. Os índices são o que torna possível usar um banco de dados espacial para grandes conjuntos de dados. Sem indexação, qualquer pesquisa de um recurso exigiria uma "varredura sequencial" de cada registro no banco de dados. A indexação acelera a

pesquisa organizando os dados em uma árvore de pesquisa que pode ser rapidamente percorrida para encontrar um registro específico[9].

O banco de dados trata as informações de projeção geográfica de maneira diferente da interface. Conversões devem ser realizadas para a correta exibição dentro dos limites do mapa projetado pelo *OpenLayers* - biblioteca de Javascript puro, gratuita, para exibição de dados em mapas, na maioria dos navegadores web modernos, sem depender do lado servidor. Essas conversões são realizadas durante a inserção e a consulta: a referência espacial no banco de dados é a EPSG:29182, já na consulta é a EPSG:4326. O Conjunto de Dados de Parâmetros Geodésicos EPSG é um conjunto de dados estruturado de Sistemas de Referência de Coordenadas e Transformações de Coordenadas, acessível através deste registro on-line (www.epsg-registry.org) ou, como arquivos zip descarregáveis, através da página inicial do EPSG em www.epsg.org. O EPSG se refere ao mercator esférico, um tipo de projeção que trata a Terra como uma esfera e é usado por grande parte dos principais provedores de APIs comerciais, como o Google Maps[6].

Os dados relativos a representação da geometria também passam por processos de conversão, que são tratados por funções próprias do PostGIS. A função *ST_GeomFromGeoJSON,* por exemplo*,* toma como entrada uma representação geojson de uma geometria e gera um objeto de geometria do PostGIS [7]. Outras funções semelhantes, abrem a possibilidade para a integração com tecnologias de inteligência de negócio geoespaciais [8], o que auxiliaria na ampliação de mecanismos para a tomada de decisões gerenciais pelos usuários do sistema.

3. Resultados e Considerações Finais

As novas consultas obtêm os pontos da sub-bacia relativos a qualidade da água e fauna utilizando o operador topológico *Contains,* que recebe como parâmetro o polígono da sub-bacia e as coordenadas do ponto [Figura 1].

```
SELECT sub_bacia.id_sub_bacia, sub_bacia.nome_bacia,
ponto_agua.id_ponto, ponto_agua.localizacao, ponto_agua.id_ponto_sia
FROM public.sub_bacia,qualidadeagua.ponto_agua
WHERE ST_Contains(sub_bacia.poligono,ponto_agua.coordenadas)
AND sub_bacia.id_sub_bacia = 3
ORDER BY id_sub_bacia, id_ponto_sia
```

Figura 1. Exemplo de consulta: pontos contidos na sub-bacia 3

Após a implementação das estruturas de consulta, conversão e armazenamento de dados espaciais, os resultados já podem ser auferidos pelo usuário do sistema. A figura 2 apresenta as melhorias observadas pelo usuário, que agora pode selecionar a sub-bacia de interesse para uma análise pontual sobre os itens espaciais associados. Observa-se no lado esquerdo as sub-bacias selecionadas e suas respectivas localizações no mapa com a cor de destaque. Verifica-se no banco de dados a evolução em relação aos atributos não espaciais. que descrevem qualitativa e quantitativamente uma entidade geográfica. A relação dos pontos com as sub-bacias foi melhorada, explorando os potenciais dos atributos espaciais, que consideram a localização e a representação do espaço geográfico usando um sistema de coordenadas. As novas consultas implementadas

estabelecem relações considerando os relacionamentos de vizinhança, ou presença dentro dos limites das sub-bacias.

Figura 2. Interface do sistema com acesso aos dados espaciais

4. Referências Bibliográficas

[1] Sabău, G. (2007) "Comparison of RDBMS, OODBMS and ORDBMS", In: Informatica Economica, Vol. XI, pp. 83-85.

[2] Singh, S. P., Singh, P. (2014) "Mapping Spatial Data on the Web Using Free and Open-Source Tools: A Prototype Implementation", In: Journal of Geographic Information System, 2014, 6, 30-39.

[3] Miranda D., Russo D., Alves R. A. L. (2012) "Upgrading to PostGIS 2.0 in the brazilian federal forensics GIS", In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXIX-B4, 2012.

[4] Manual do PostGIS (2016). Disponível em: <http://postgis.net/docs/manual-2.0/>. Acesso em 3 de setembro de 2016.

[5] SRIDs (Spatial Reference Identifiers). Disponível em: <https://msdn.microsoft.com/pt-br/library/bb964707.aspx>. Acesso em 10 de novembro de 2016.

[6] Spherical Mercator (2016). Disponível em: <http://docs.openlayers.org/library/spherical_mercator.html>. Acesso em 12 de novembro de 2016.

[7] ST_GeomFromGeoJSON (2016). Disponível em: <http://www.postgis.org/docs/ST_GeomFromGeoJSON.html>. Acesso em 15 de novembro de 2016.

[8] Enabling Geospatial Business Intelligence (2009). Disponível em: <https://timreview.ca/article/289>. Acesso em 16 de novembro de 2016.

[9] ELMASRI, Ramez; NAVATHE, Shamkant. Sistemas de Bancos de Dados. 4. ed. São Paulo: Pearson, 2005.

Mapeamento de Padrões de Acidentes de Trânsito com Vítimas Fatais a partir de Dados Públicos do Governo do Estado do Rio Grande do Sul

Jorge Alberto F. Flores. Jr[1], Leonardo C. Steffenello[1], Ana T. Winck[1]

[1]Laboratório de Computação Aplicada (LaCa)
Universidade Federal de Santa Maria – Santa Maria – RS – Brazil

{jjunior,lsteffenello,ana}@inf.ufsm.br

Abstract. *Every year, statistics demonstrate an increasingly growth about fatality in Rio Grande do Sul traffic. This paper shows a study mapping traffic fatal accident patterns occurred in Rio Grande do Sul and in Santa Maria from 2007 to February 2016, in order to identify their main factors. To do so, we applied a Knowledge Discovery in Databases process, by means of the execution of Apriori and J48 algorithms in the WEKA tool. We employed a methodology with a extensive preprocessing, algorithm execution and a particular analysis of the generated rules for knowledge mapping about fatal traffic accidents in Rio Grande do Sul.*

Resumo. *A cada ano, estatísticas demonstram um aumento da fatalidade no trânsito do Rio Grande do Sul. Este trabalho apresenta um estudo para mapear os padrões de acidentes de trânsito com vítimas fatais ocorridos no Rio Grande do Sul e em Santa Maria, entre 2007 e fevereiro de 2016, a fim de identificar os principais fatores envolvidos. Para isso, é empregado um processo de Descoberta de Conhecimento em Bases de Dados, por meio da execução dos algoritmos Apriori e J48, na ferramenta de mineração de dados WEKA. É apresentada uma metodologia de extenso pré-processamento dos dados, execuções dos algoritmos e análises das regras geradas para um mapeamento dos conhecimentos adquiridos acerca dos acidentes de trânsito.*

1. Introdução

A Secretaria de Transportes e o Departamento Estadual de Trânsito do Rio Grande do Sul (DETRAN RS), dentre outras várias atribuições, atuam, em conjunto com outros órgãos, estabelecendo a política de transportes e fiscalizando o trânsito de veículos terrestres no Rio Grande do Sul (RS). Com o objetivo de informar e conscientizar a sociedade, o governo do estado do RS, através do Portal de Dados Abertos do RS (2016), disponibilizou uma base de dados, produzida pelo DETRAN RS, da relação de acidentes de trânsito com vítimas fatais no estado. Essa base de dados, denominada de Crimes de Trânsito, é constituída de 16.016 registros e 18 atributos, na qual cada registro representa um acidente de trânsito com vítima fatal ocorrido no estado entre o período de 2007 e fevereiro de 2016, e cada atributo indica uma característica envolvida nestes acidentes.

A W3C (2010) define dados abertos governamentais como os dados produzidos pelo governo e colocados à disposição das pessoas. Entretanto, um dos grandes desafios, em se tratando de dados abertos, é processar, analisar e tirar conclusões desse

imenso volume de dados brutos. Isso evidencia a necessidade de utilizar técnicas que facilitem a tarefa de extração de conhecimento, o que pode ser realizado por meio do processo de Descoberta de Conhecimento em Bases de Dados (KDD). Fayyad et al. (1996) definem o KDD como uma sequência de passos interativos e iterativos de apoio à tomada de decisão, fundamentais quando da existência de grandes volumes de dados e envolvendo as etapas de Seleção, Pré-Processamento, Transformação, Mineração de Dados e Interpretação. Diante da problemática do trânsito e da divulgação pública de seus dados, este trabalho tem como objetivo principal aplicar o processo de KDD a fim de identificar padrões sobre os dados referentes a acidentes de trânsito com vítimas fatais no RS, entre os anos de 2007 e fevereiro de 2016.

2. Metodologia

Para atender aos objetivos deste trabalho, foi aplicada uma sequência de passos fundamentais. Primeiramente, foi realizado o download do conjunto de dados Acidentes de Trânsito com Vítimas Fatais no RS. Os mesmos foram analisados, visando o entendimento dos registros e atributos. Após a seleção dos dados, foi possível aplicar as etapas de pré-processamento e transformação visando a adequação dos dados aos algoritmos a serem executados. Tratada a base de dados, aplicaram-se os algoritmos de mineração a fim de compreender os fatores que contribuem com a acidentalidade fatal no trânsito do RS. Para tanto, optou-se em utilizar regras de associação e árvores de decisão. Estas etapas foram repetidas, até que se chegasse a conhecimentos relevantes.

Neste trabalho, a etapa de Pré-Processamento se preocupou com que os resultados da mineração não fossem afetados pela grande quantidade de dados errôneos, irrelevantes e inconsistentes da base selecionada. Por isso, todos os seus dados e seus atributos tiveram de serem analisados e, quando necessário, aplicados a técnicas de eliminação e limpeza. Como consequência dessa etapa, foram criados novos atributos que possam auxiliar na descoberta de conhecimentos, excluídos atributos não relevantes ou simplesmente modificado a maneira como se apresentavam. Em relação aos dados, quando preenchidos de forma ruidosa, tiveram de serem adequados para se tornarem representativos. O pré-processamento fez com que os dados ruidosos fossem tratados de forma que não afetem os resultados da mineração.

Na execução da etapa de Mineração de Dados optou-se por utilizar as técnicas de regras de associação e árvores de decisão, por meio dos algoritmos Apriori (Agrawal et al., 1996) e J48 (Quinlan, 1993). Com a finalidade de encontrar resultados relevantes ao problema do tráfego, executaram-se testes visando à descoberta de padrões de acidentes em todo o RS e também apenas para Santa Maria. Para isso, foram executados diversos testes com diferentes configurações de parâmetros, através dos dois algoritmos, com diferentes valores de suporte e confiança e com e sem a presença de atributo classe.

Realizadas as execuções dos casos de testes, o último passo do processo de KDD envolveu a interpretação dos resultados gerados da aplicação dos algoritmos, a fim de transformá-los em conhecimentos úteis aos objetivos deste trabalho. A interpretação e avaliação dos resultados foram executadas por meio da análise das regras geradas pelo algoritmo Apriori e das árvores geradas pelo algoritmo J48, destacando os resultados mais relevantes e, principalmente, realizando uma comparação entre os dois algoritmos. Dessa forma, espera-se facilitar o entendimento sobre a acidentalidade fatal no estado.

3. Resultados

Com a execução dos algoritmos Apriori e J48 e pela interpretação de seus resultados, pode-se perceber que a descoberta dos melhores resultados depende muito do pré-processamento aplicado e da qualidade dos dados utilizados e que o número de regras geradas é sempre inversamente proporcional aos valores determinados para o suporte e a confiança. Além disso, devido à má distribuição dos registros entre as categorias, a maior parte das regras geradas continha sempre as mesmas características envolvidas. Entretanto, removendo-se da base de dados os atributos cujos registros foram considerados mal distribuídos, conseguiu-se chegar a resultados mais relevantes e que exploraram melhor os demais atributos. A aplicação da mineração com a presença de atributo classe também auxiliou na geração de regras específicas para os atributos.

Como foram gerados mais de 5000 resultados para regras de associação e árvores de decisão, não foi possível exibir todos os resultados encontrados. Nesse sentido, as regras produzidas pelos algoritmos são destacadas em termos da interpretação daquelas que melhor contribuíram com os objetivos deste trabalho. Dentre as mais relevantes, destacam-se:

a) Tanto para o RS, quanto para Santa Maria, há uma maior propensão à ocorrência de acidentes com vítimas fatais nos finais de semana e no período noturno;

b) No RS, apresenta-se uma maior propensão a acidentes em vias municipais. Caso se leve em comparação apenas as estradas rodoviárias, ocorre com maior frequência acidentes em vias estaduais do que em vias federais. Em Santa Maria, o panorama é diferente: apresenta-se uma maior propensão a acidentes em vias federais;

c) Para ambas regiões de estudo, há uma maior tendência à ocorrência de atropelamentos em vias municipais e de colisões em rodovias;

d) No RS, percebe-se uma maior propensão a acidentes na região metropolitana do estado. Quando analisado o atributo ‘Logradouro’ da base de dados do município de Santa Maria, percebe-se uma maior propensão a acidentes na BR-287;

e) Tanto para o RS quanto para Santa Maria, há uma maior tendência a colisões. Entretanto, se considerar apenas as vias municipais, há um predomínio de atropelamentos;

f) Os atropelamentos durante a madrugada, por ser uma ocorrência onde o pedestre é a principal vítima e onde existe um menor fluxo de pessoas no trânsito, aumentam a probabilidade de resultarem em óbitos;

g) Os acidentes considerados como homicídio doloso durante a madrugada tendem a ocasionar em fuga do local do crime por parte dos motoristas causadores;

h) Motoristas embriagados se envolvem principalmente em ocorrências de capotagens e choques a objetos fixos. Na maioria dos casos, os crimes são considerados como homicídio doloso, quando assume-se o risco de causar mortes; e

i) Motoristas sem CNH, independente do tipo do acidente, tendem a fugir do local do crime, e também causam crimes considerados como homicídio doloso.

Realizando uma análise comparativa entre os algoritmos utilizados, ambos se mostraram eficazes na descoberta de conhecimento. Interpretando as descobertas feitas neste trabalho podemos concluir que eles se complementam, e, apesar apresentarem os

resultados de maneira diferente, ambos os algoritmos podem ser utilizados em conjunto a fim de comprovar os resultados alcançados.

4. Conclusão

Para o desenvolvimento deste trabalho foi aplicado o processo de KDD sobre dados de ocorrências de acidentes de trânsito com vítimas fatais no RS, entre os anos de 2007 e fevereiro de 2016, a fim de analisar os padrões encontrados e realizar um mapeamento dos conhecimentos relevantes. Esse estudo ocorreu por meio da geração de regras de associação e árvores de decisão, utilizando os algoritmos Apriori e J48, na ferramenta de mineração de dados WEKA.

As técnicas e estratégias adotadas, bem como a conclusão sobre cada um dos experimentos realizados foram sendo registrados e documentados. Dessa forma, foi possível apresentar um mapeamento dos conhecimentos alcançados a fim de melhorar o entendimento sobre os resultados. Vale ressaltar a boa iniciativa do governo do estado, em conjunto com o DETRAN RS, de tornar estas informações públicas. Porém, pode-se destacar como ponto negativo, a qualidade de preenchimento dos seus dados, muitas vezes, com erros, faltantes e inconsistentes. Apesar disso, o processo de KDD foi realizado e atingiu os objetivos esperados com a execução deste trabalho.

Em suma, este trabalho visa se apresentar como uma contribuição na demonstração da viabilidade de utilização dos conceitos de mineração de dados a fim de analisar um problema tradicional. Além disso, espera-se que os resultados apresentados neste trabalho possam orientar novos estudos sobre o tema ou assuntos relacionados.

Referências

Agrawal, R. et al. (1993). A Mining Association Rules Between Sets of Items in Large Databases. In: ACM SIGMOD International Conference on Management of Data. 2nd edition

Fayyad, U. M. et al. (1996). Advances in Knowledge Discovery and Data Mining. In: Emerald Group Publishing. 1st edition.

Portal de Dados Abertos do Rio Grande do Sul. “Dados Abertos de seu Interesse”, http://dados.rs.gov.br, 18 abr. 2016.

Quinlan, J.R. (1993) C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers.

W3C. “Manual dos Dados Abertos: Governo do Rio Grande do Sul”, http://www.w3c.br/pub/Materiais/PublicacoesW3C/Manual_Dados_Abertos_WEB.pdf, 18 abr. 2016.

Ferramenta de Modelagem de Bancos de Dados Relacionais brModelo v3

Carlos Henrique Candido[1], Ronaldo dos Santos Mello[2]

[1]TRE-MT – Tribunal Regional Eleitoral de Mato Grosso
Avenida Historiador Rubens de Mendonça, 4750, Cuiaba– MT – Brasil

[2]Departamento de Informática e Estatística (INE) – Universidade Federal de Santa Catarina (UFSC) - Caixa Postal 476 – 88.010-970 – Florianópolis – SC – Brasil

chcandido@hotmail.com, r.mello@ufsc.br

Abstract. *This article describes the features and announces the beta release of the third version (v3) of the Brazilian database relational modeling tool, called brModelo. This tool is used in most of the brazilian educational institutions that offer courses in information technology area, including technical courses, degree coursers and university graduate courses. His great contribution is related to the teaching of Database design.*

Resumo. *Este artigo descreve as funcionalidades e anuncia o lançamento da terceira versão (v3) da ferramenta de modelagem relacional de banco de dados, brModelo. Esta ferramenta é utilizada na maioria das instituições de ensino brasileira que oferecem cursos na área de tecnologia da informação, incluindo cursos técnicos e profissionalizantes, de graduação e de pós-graduação. Sua grande contribuição é percebida no campo do ensino de projeto de bancos de dados relacionais.*

1. Introdução

Em 2005 foi desenvolvida uma ferramenta de código aberto e totalmente gratuita voltada para ensino de modelagem de banco de dados relacionais com base na metodologia defendida por Carlos A. Heuser no livro "Projeto de Banco de Dados". Esta ferramenta foi concebida como trabalho de conclusão do curso de especialização em banco de dados pelas universidades UFSC (SC) e UNIVAG (MT), orientado pelo Professor Dr. Ronaldo dos Santos Mello, após se constatar a inexistência de uma ferramenta nacional que pudesse ser utilizada para essa finalidade.

Em junho de 2006 foi realizado o lançamento da versão 2.0 da ferramenta. Naquela época, acreditava-se que a modelagem de bancos de dados relacionais poderia ser substituída nos próximos dez anos por ferramentas de desenvolvimento de software orientadas a objeto que propunham a persistência do modelo de classes diretamente em um sistema de gerência de banco de dados (SGBD), sem, portanto, a necessidade de uma análise do modelo voltada exclusivamente para os dados.

Em 2015, passados dez anos de disponibilização da primeira versão da ferramenta, verifica-se ainda a existência de grande demanda por conhecimentos na área de modelagem de dados relacionais e, inclusive, a produção de novos artefatos para

abstração dos diagramas de dados, em especial, os conceitos relacionados à modelagem conceitual e lógica. O mercado não substituiu os SGBDs relacionais e novas funcionalidades têm sido implementadas. Também por isso, a brModelo continua a ser utilizada em várias universidades e centros de ensinos técnicos no pais e até mesmo no exterior.

Tudo isso motivou-nos a continuar os trabalhos de desenvolvimento e a publicar uma nova versão do brModelo, também baseada em código aberto, porém em linguagem de programação mais atual e com possibilidade de trabalhos colaborativos, cujo beta pretende ser lançada no site da ferramenta na data da próxima Escola Regional de Banco de Dados, oportunidade onde serão colhidas sugestões de melhoria pelo público para análise e possíveis aprimoramentos.

2. Melhorias Introduzidas na Nova Versão da brModelo

A nova versão está sendo desenvolvida em Java ™, mantém todas as funcionalidades da versão anterior e pretende implementar algumas modificações defendidas por alguns professores de modelagem de bancos de dados relacionais, publicadas em sites sobre o assunto na Internet. Assim sendo, ela oferecerá uma nova oportunidade para avanços no ensino de banco de dados.

Além disso, esta nova versão da brModelo implementa outras notações diagramáticas que podem ser uteis no processo de modelagem conceitual. Desta forma, suas bases poderão ser utilizadas para o desenvolvimento de outras notações e será usada no projeto brUML (ferramenta para o ensino de UML, ainda em fase de levantamento de requisitos).

No campo das novas funcionalidades, esta nova versão da brModelo apresenta uma nova interface com o usuário (incluindo recursos tradicionais, como copiar/colar, zoom, teclas de atalho e etc.), um novo padrão de codificação baseado no *Code Conventions for the Java Programming Language*, com ênfase nos padrões de projeto (principalmente *Factory* e *Strategy*), internacionalização, ajuda interativa e extensão do diagrama originalmente proposto pelo Dr. Heuser (2001), baseado no trabalho de Peter Chen (1990), com a inclusão de união de entidades.

3. Apresentação da Nova Versão da Ferramenta

Esta seção apresenta a nova versão da ferramenta brModelo (v3 – beta 1), que ainda em 2017 substituirá a ferramenta atualmente em uso (brModelo 2.0, disponível no site do autor: www.sis4.com).

O grande diferencial da ferramenta, quando comparada às demais, é o fato dela ter sido criada com foco no ensino e na aprendizagem da modelagem de dados relacional em nível técnico e acadêmico, ao contrário daquelas voltadas exclusivamente para auxiliar no trabalho dos profissionais de desenvolvimento de aplicações de banco de dados. A nova versão mantém o mesmo foco e aperfeiçoa os novos conceitos na área afim.

O assunto é relevante para a comunidade que atua na área de banco de dados relacionais, principalmente no tocante ao ensino e aprendizado das técnicas de modelagem. Prova disso é que foram realizados mais de 500.000 (quinhentos mil) downloads da ferramenta desde sua publicação.

Do volume de downloads, cerca de trezentos mil foram contabilizados apenas no site Baixaki (BAIXAKI, 2017), sessenta e sete mil no site ZIGG OUL (FERRAMENTA, 2017), onze mil no site Programas & jogos (PROGRAMA, 2017), além de outros, conforme mostra a Figura 1.

Figura 1. Quantidade de downloads.

As modificações realizadas na interface da ferramenta (Figura 2) foram inspiradas nos conceitos de usabilidade defendidos, entre outros, por *freedesktop.org*. Isso contribui para uma melhor experiência de uso.

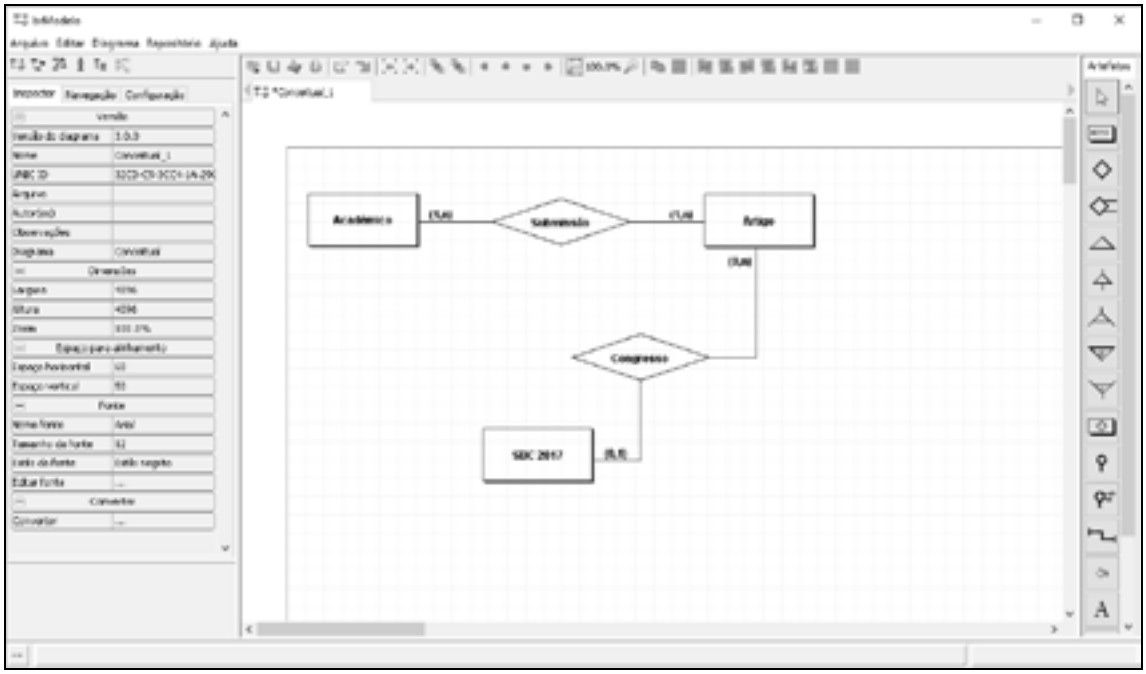

Figura 2. Imagem da tela principal da nova versão da ferramenta brModelo.

Referências

CHEN, Peter. Modelagem de Dados: A Abordagem Entidade-Relacionamento para Projeto Lógico; Tradução Cecília Camargo Bartalotti. São Paulo, McGraw-Hill, 1990.

BAIXAKI - ferramenta para criar modelos relacionais de um banco de dados roda direto do pendrive. Baixaki. Disponível em: <http://www.baixaki.com.br /download/brmodelo.htm#ixzz4baRJ99Ml>. Acessado em 10 fev. 2017.

HEUSER, Carlos Alberto. Projeto de Banco de Dados, Porto Alegre: Instituto de informática da UFRGS, Sagra Luzzato, 2001. Série livros didáticos n.º 4.

FERRAMENTA gratuita voltada para o ensino de modelagem em banco de dados relacional. Zigg UOL. Disponível em <http://ziggi.uol.com.br/downloads/brmodelo>. Acessado em 10 fev. 2017.

PROGRAMA grátis para criar objetos e esquemas a partir de bases de dados. Programas & jogos. Disponível em <https://brmodelo.programasejogos.com>. Acessado em 10 fev. 2017.

Estudo comparativo entre sistemas de gerenciamento de bancos de dados relacionais e não relacionais para o armazenamento e busca de metadadados MARC

Jader Osvino Fiegenbaum[1], Evandro Franzen[1]
[1]Centro Universitário Univates
Rua Avelino Tallini, 171, Bairro Universitário – Lajeado – RS – Brasil

Abstract. *The use of archive management system reduces the labor and cost of internal libraries. These systems mostly need to store the data in a metadata standard for dealing with flexible and dynamic information such as MARC. Traditionally relational databases are used, most of which have rigid structures for data storage. The purpose of this study was to compare the performance of DBMS's PostgreSQL and MongoDB for storage and search MARC metadata.*

Resumo. *A utilização de sistemas de gestão de acervo reduz o trabalho e o custo interno das bibliotecas. Estes sistemas, em sua maioria necessitam armazenar os dados em algum padrão de metadados para lidar com informações flexíveis e dinâmicas como o MARC. Tradicionalmente são usados bancos de dados relacionais, que em sua maioria apresentam estruturas rígidas para o armazenamento de dados. O propósito deste estudo foi comparar a performance dos SGDB PostgreSQL e MongoDB para o armazenamento e busca de metadados MARC.*

1. Introdução

O processo de automação de bibliotecas é indispensável para o atendimento de demandas informacionais, principalmente em bibliotecas universitárias, pois os usuários necessitam ter acesso à informação de diversas áreas. O uso de softwares para o gerenciamento de acervos bibliográficos é um fator decisivo no desempenho das funções da biblioteca.

Para suprir as necessidades destes sistemas, em 1960 foi criado o padrão de metadados MARC (*Machine Readable Cataloging)* para armazenar registros bibliográficos de diferentes tipos de objetos, com o objetivo de automatizar o processo de catalogação (FURRIE ,2003). Gil-Leiva (2007) e Furrie (2003), definem o padrão MARC como um conjunto de números, letras e símbolos combinados e adicionados aos registros catalográficos.

Um registro bibliográfico que segue o formato MARC possui dados encapsulados em campos e subcampos, que podem ser entendidos como *tags*. Cada *tag* permite organizar e gerir a informação, além de permitir a sua recuperação. Portanto, diferentes tipos de objetos não serão catalogados da mesma forma, não contendo os mesmos conjuntos de campos e subcampos, uma vez que possuem estrutura diferente.

Em função da estrutura dinâmica, o armazenamento de tais informações não se adapta tão facilmente ao modelo relacional, que apresenta uma estrutura rígida, baseada em tabelas e campos. A busca por sistemas com estruturas mais flexíveis levou o

desenvolvimento de bancos NoSQL (Not Only SQL), que possuem como principal característica armazenar dados com estrutura flexível e tratamento de grandes volumes de dados (Lóscio, Oliveira e Pontes ,2011).

Os estudos sobre modelos NoSQL tem se tornado cada vez mais frequentes, principalmente com o foco na comparação (POLITOWSKI, MARAN, 2014) e interoperabilidade com modelos relacionais (SCHREINER, 2015) ou ainda como alternativa para o armazenamento e busca de dados não convencionais (SCHULZ, 2016).

O fato de registros bibliográficos em MARC serem dinâmicos e semiestruturados possibilita que os mesmos possam ser armazenados e processados em bancos NoSQL. A principal questão a ser investigada é se o armazenamento em um modelo flexível, como NoSQL apresenta um desempenho compatível ou superior aos modelos tradicionais e, desta forma, pode ser uma alternativa para armazenar estes tipos de dados.

2. Procedimentos metodológicos

A metodologia usada neste trabalho foi de natureza quantitativa, pois envolveu a coleta de dados relacionados ao desempenho de sistemas que implementam os modelos relacional e NoSQL em consultas que utilizam dados no formato MARC. Os sistemas de gerência de bancos de dados escolhidos foram o PostGreSQL (relacional) e o MongoDB (orientado a documentos). Ambos possuem código aberto e estão entre os sistemas mais utilizados no cenário atual.

A modelagem de dados relacional inclui duas tabelas para controlar as *tags* e os valores para cada material. Neste caso para recuperar ou atualizar os dados é necessário acessar a tabela de cadastro do material e as tabelas de definem as informações no formato MARC. O armazenamento no MongoDB envolveu somente uma coleção, com estrutura flexível, para manter as informações relacionadas ao padrão. Para execução dos testes, foi desenvolvido um protótipo que contempla testes de busca, tolerância a falhas, carga de dados, stress e volume do banco de dados (Figura 1).

Figura 1 – Interface do protótipo para execução dos testes

O teste de busca teve como objetivo determinar o desempenho de cada SGBD (Sistema de Gerenciamento de banco de dados) ao lidar com a busca de materiais. O teste consistiu em medir o tempo ao buscar materiais cadastrados, combinando campos MARC nos filtros. Uma das estratégias foi analisar o desempenho de consultas que

retornam dados e a segunda é a execução de consultas que apenas contam registros.

O teste de volume buscou verificar a desempenho do sistema ao lidar com grande volume de dados em relação ao tempo. Foram executadas consultas que retornaram todos os materiais armazenados, e foram considerados os tempos de execução das consultas e não da obtenção dos dados, a partir dos recursos ou cursores retornados em cada caso.

3. Resultados e discussão

Nesta seção apresenta os resultados obtidos nos testes de desempenho de busca e volume. A execução dos testes de busca foi realizada de duas maneiras, a primeira exibe os registros retornados e a segunda exibe somente a quantidade de registros e tempo de execução da consulta em ambos SGDB's. Em ambos os casos, foi possível combinar filtros através de seleção de campos e operadores booleanos (e/ou). A Tabela 1 apresenta os resultados.

Tabela 1 – Resultado do teste de busca

Condição	Quantidade de registros	Postgres (exibe registros)	Postgres (conta registros)	Mongo (exibe registros)	Mongo (conta registros)
CONTEM 'BANCO' E 650.a CONTEM 'INFORMATICA' E 100.a CONTEM 'el'	10000	0,489892s	0,446063s	1,541534s	0,232550s
CONTEM 'BANCO' E 650.a CONTEM 'INFORMATICA' OU 100.a CONTEM 'el'	40000	1,001012s	0,831537s	5,965069s	0,458339s
CONTEM 'SISTEMAS' E 260.a CONTEM 'SAO PAULO' E 650.a CONTEM 'banco'	20000	2,870094s	0,613667s	2,984691s	0,381753s
CONTEM 'SISTEMAS' E 260.a CONTEM 'RIO' E 650.a CONTEM 'banco'	10000	0,490941s	0,451199s	1,734503s	0,386137s
CONTEM 'SISTEMAS' E 260.a CONTEM 'RIO' OU 650.a CONTEM 'banco' E 100.a CONTEM 'Garcia'	10000	0,505619s	0,472283s	1,571384s	0,267236s

Os resultados apontam uma performance superior do MongoDB em relação ao PostgreSQL quando utilizada a interface que não exibe os registros, o que demonstra a performance do banco de dados. Porém, a exibição dos registros indica que o PostgreSQL possui performance superior no contexto da aplicação. Isso ocorreu porque no caso da exibição de registros são retornados todos os resultados do *resource* e do cursor. Iterar sobre estes dados no PostgreSQL é mais rápido do que iterar sobre todo o cursor do MongoDB. A diferença entre eles aumenta proporcionalmente de acordo com a quantidade de registros.

Tabela 2 – Resultado do teste de volume

Registros	Tempo Postgres (segundos)	Tempo MongoDB (segundos)
21	0,00066363	0,00004863
2100	0,00277275	0,00004775
21000	0,02068025	0,00004663
210000	0,19547763	0,00005400
525000	0,45658263	0,00004738
1050000	0,93997550	0,00005313

Observa-se que o sistema relacional obtém dados armazenados na memória RAM, e o segundo mantém uma conexão aberta com o banco de dados. O MongoDB foi concebido para grande volume de dados, na ordem de terabytes de informação. Nesse cenário seria impossível o sistema retornar dados sem utilizar um cursor, pois não haveria memória disponível para isso.

No teste de volume foram executadas consultas que retornam todos os materiais armazenados. Os resultados são apresentados na Tabela 2.

Os resultados apontam uma performance superior do padrão NoSQL. É possível perceber que o tempo do banco relacional aumenta proporcionalmente em relação a quantidade de registros. O mesmo não ocorre para o MongoDB, devido ao fato deste utilizar memória virtual e também ter sido executado logo após o teste de carga. Portanto, é provável que todos os dados ainda estivessem em memória RAM.

4. Conclusão e trabalhos futuros

Este trabalho apresentou uma análise comparativa de SGBDs, com ênfase no desempenho demonstrado em consultas. Embora durante as pesquisas tenham sido realizados também testes de carga, stress e tolerância a falhas, optou-se neste artigo por apresentar somente os resultados das consultas.

Observou-se que o modelo NoSQL é uma alternativa promissora, pois exibe um bom desempenho na execução das consultas. Entretanto, é necessário criar alternativas que melhorem os resultados na aplicação, na exibição dos dados pela interface. O teste de volume demonstra que o MongoDB possui performance superior ao PostgreSQL, isso se deve principalmente ao fato do registro bibliográfico ser armazenado em apenas um documento, enquanto para armazenar um registro completo com todas as etiquetas MARC no PostgreSQL são necessários múltiplos registros, em tabelas diferentes.

Trabalhos futuros apontam para a necessidade de avaliar outras estruturas de armazenamento, como grafos, colunas e o desenvolvimento de aplicações que permitam testes reais, com usuários e dados existentes em sistemas de gestão de bibliotecas.

Referências

FURRIE, B. **Understanding MARC bibliographic: machine-readable cataloging**. 7 ed. rev. Washington, D. C.: Library of Congress; Follet Software, 2003.

GIL-LEIVA, Isidoro. **A indexação na internet**. Brazilian Journal of Information Science. v.1, n.2, p.47-68. 2007.

LÓSCIO, Bernadette Farias; OLIVEIRA, Hélio Rodrigues de; PONTES, César de Sousa. **NoSQL no desenvolvimento de aplicações Web colaborativas**. Simpósio Brasileiro de Sistemas Colaborativos, 2011. Paraty. 2011.

POLITOWSKI, Cristino; MARAN, Vinícius. **Comparação de Performance entre PostgreSQL e MongoDB**. ERBD 2014, São Francisco do Sul, 2014.

SCHREINER, Geomar A.; DUARTE, Denio; DOS SANTOS MELLO, Ronaldo. **Análise de Abordagens para Interoperabilidade entre Bancos de Dados Relacionais e Bancos de Dados NoSQL**. ERBD 2015. Caxias do Sul, RS. 2015.

SCHULZ, Wade L. et al**. Evaluation of relational and NoSQL database architectures to manage genomic annotations**. Journal of Biomedical Informatics, v. 64, p. 288-295, 2016.

Aplicação da Análise de Sentimentos em Frases das Redes Sociais sobre Empresas de Serviços de Telecomunicação

Elvis Kesley de Assis[1], Renata L. Rosa[1], Demóstenes Z. Rodríguez[1], Rosangela de Fátima Pereira[2], Tereza Cristina Melo de Brito Carvalho[2], Graça Bressan[2]

[1]Departamento de Ciência da Computação– Universidade Federal de Lavras (UFLA)
Caixa Postal 37200-000 – Lavras – MG – Brasil

[2]Departamento de Engenharia da Computação – Universidade de São Paulo
São Paulo, Brasil.

eassis@sistemas.ufla.br,{demostenes.zegarra,renata.rosa}@dcc.ufla.br,
{rpereira, carvalho, gbressan}@larc.usp.br

Abstract. *This article performs an analysis of texts extracted from the social network, Twitter, in relation to the telecommunication services topics offered by four companies, applying the sentiment analyses to detect complaints and dissatisfaction of users of a particular service. The study also makes a relation between signal quality complains extracted on social network with the number of base stations of the user's social network region, which is extracted by a mobile service application. The main topics of the complaints are detected and a monitoring system points out the main problems to the telecommunication companies and national regulatory agencies.*

Resumo. *Este artigo faz a análise de textos extraídos da rede social, Twitter, em relação a tópicos de serviços de telecomunicação ofertados por quatro empresas, efetuando a análise de sentimentos para detecção de reclamações ou insatisfação de usuários sobre um determinado serviço. O estudo também faz uma relação entre qualidade de sinal e o número de estações bases da região do usuário da rede social. Os principais objetos da reclamação são detectados e um sistema de monitoramento aponta os principais problemas às empresas de telecomunicações e agências nacionais reguladoras.*

1. Introdução

As redes sociais servem como um rico repositório de dados, onde o consumidor compartilha as suas experiências positivas e negativas sobre produtos e serviços.

A extração da opinião do usuário pode ser efetuada pela análise de sentimentos de frases coletadas. Conhecendo o sentimento do usuário sobre um determinado produto ou serviço, pode ser feita uma sugestão de tópicos relacionados à opinião do usuário por meio de um sistema de recomendação (SR). A análise de sentimentos como a análise afetiva podem melhorar o desempenho de um SR [Rosa et al. 2015]. Os sistemas de monitoramentos também podem fazer uso da análise de sentimentos e podem ser utilizados em aplicações em geral, desde saúde até o monitoramento da qualidade de um sinal de celular [Rodriguez et al. 2015]. Porém, existem poucos trabalhos sobre monitoramento de qualidade de chamadas de celulares e estudos abordando quais os motivos da baixa qualidade de uma chamada.

Com o objetivo de filtrar quais dados extraídos das redes sociais devem ser armazenados e quais devem ser desconsiderados, foi desenvolvido um sistema de análise de sentimentos para contabilizar as reclamações e demais relatos negativos dos usuários e consumidores em relação a quatro empresas de telecomunicação no Brasil. Dos dados coletados, foram extraídos os principais objetos de cada reclamação e as frases relacionadas a uma qualidade de sinal ruim tiveram sua localização geográfica extraída para comparar com a quantidade de estações bases encontrada na mesma região geográfica, de acordo com o resultado de um aplicativo de serviço móvel para averiguação dos resultados de qualidade de sinal. Este artigo apresenta estudos sobre as principais insatisfações dos usuários sobre serviços de telecomunicação referentes a quatro empresas principais, objeto da reclamação e localização geográfica extraída do usuário de uma rede social utilizando a análise de sentimentos. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 mostra alguns resultados preliminares. Finalmente, a Seção 4 apresenta as conclusões finais e trabalhos futuros deste projeto em andamento.

2. Trabalhos Relacionados

As informações disponíveis na Internet são várias e a mineração de dados torna-se necessária para a melhor utilização desses dados. A análise de sentimentos também auxilia na mineração de dados na seleção de opiniões positivas, negativas ou neutras.

Medir a polaridade de sentimentos em textos escritos por usuários de um serviço é uma prática comum [Turney 2002]. De acordo com as pesquisas de análise de sentimento, é possível utilizar várias técnicas para avaliar a intensidade de sentimentos de uma frase. Uma das técnicas é por meio de um dicionário de palavras ou análise léxica; o dicionário WordNet [Turney 2002] é formado por palavras estáticas para a análise de sentimentos, porém não considera gírias e *emoticons*. Outros dicionários estáticos são SentiStrength e Sentimeter-Br [Rosa et al. 2013] que possui suporte ao idioma português – Brasil.

Os dispositivos móveis celulares têm o seu uso massivo atualmente, e portanto, vê-se necessário analisar o nível de satisfação de seus clientes perante os seus dispositivos e as empresas de telecomunicação que prestam serviços ao cliente. Trabalhos de coleta de dados a respeito de serviços de telecomunicação ainda são escassos [Jony et al. 2015][Zheng et al. 2016], porém importantes, pois Big Data oferece uma infinidade de oportunidades para as operadoras de rede móvel melhorar a qualidade de serviço ao usuário.

O aplicativo (APP) Anatel Serviço Móvel[1] possibilita verificar a localização das estações bases por região geográfica, um histórico de 12 meses de indicadores de qualidade, além do ranking de prestadoras de serviços de comunicações móveis. Para verificar se as reclamações dos usuários poderiam estar relacionadas com a presença escassa de estações bases, as frases extraídas da rede social que continham a região geográfica preenchida foram relacionadas com os resultados do aplicativo.

3. Experimentos

Foram coletadas 4050 frases da rede social Twitter, com as palavras chaves “problema”, “reclamação” e “horrível”, seguidas do nome da empresa de telecomunicação. Assim,

[1] http://gatewaysiec.anatel.gov.br/mobileanatel - Visitado em 25/11/2016

cada empresa teve em torno de 1012 ou 1013 tweets analisados no período de 1 mês. As frases coletadas foram analisadas por um script automático que efetuava a soma da intensidade final de sentimentos, por análise léxica, e foram extraídos os parâmetros mostrados na Figura 1.

Figura 1. Resultado de extração de frases da rede social.

Na análise léxica utilizada as palavras positivas e negativas foram pontuadas de +1 a +5 ou -1 a -5. Foram desconsiderados textos com polaridade neutra de sentimentos e consideramos somente os textos de polaridade negativa. Dessa forma, a base de dados a ser armazenada é reduzida e otimizada para o cenário de captura de insatisfações e reclamações. Observe que, na primeira frase apresentada na Figura 1, o usuário da rede social cita um problema com a Internet e a região de sua localização, Cascavel - Paraná; na segunda frase a pessoa está com problemas de sinal de comunicação e a sua localização é mostrada como sendo de Salvador, Bahia.

Figura 2. Resultado das estações bases localizadas em Salvador – BA segundo Aplicativo da Anatel.

Relacionamos os resultados das frases contendo reclamações com os resultados obtidos do aplicativo que mostra as estações bases da região geográfica do usuário da rede social, como a região de Salvador – Bahia, conforme mostra a Figura 2.

Por meio do aplicativo é possível verificar que muitas vezes quando o usuário da rede social se mostra insatisfeito com o sinal fraco de celular é porque a empresa portadora do sinal possui poucas estações bases na região.

4. Considerações Finais e Trabalhos Futuros

Os resultados mostraram que 70% das reclamações são devidas a pouca existência de estações bases na região do usuário, tal fato foi comprovado relacionando manualmente a escassez das estações bases citado na plataforma da ANATEL com as regiões de reclamação coletadas no Twitter. Para melhores resultados seria preciso extrair o bairro no qual o usuário da rede social está situado em vez do estado ou capital. Pois, o detalhamento do Aplicativo Serviço Móvel é a nível de cidades. Outros 30% das reclamações estão relacionadas a outros temas de insatisfações, tais como aparelhos celulares ou preço de tarifas oferecido pelas prestadoras de serviços de comunicação.

Como trabalho futuro pretende-se aumentar a quantidades de palavras chaves a serem analisadas, gerar um sistema de monitoramento de mensagens que constem os principais tópicos de insatisfação para envio às empresas que oferecem tais serviços, assim como também pode ser enviado à agência reguladora, Agência Nacional de Telecomunicações (ANATEL) e ao usuário final.

Referências

Jony, R. I., Habib, A., Mohammed, N. and Rony, R. I., (2015) "Big Data Use Case Domains for Telecom Operators," *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, Chengdu, p. 850-855.

Rodriguez, D. Z., Souza, J. M., Pivaro, G. F. (2015) "Apparatus and method for evaluating voice quality in a mobile network" US. Patent number US 9,078,143 B2 by US Patent and Trademark Office.

Rosa, R. L., Rodriguez, D. Z. and Bressan, G., (2013), "SentiMeter-Br: A Social Web Analysis Tool to Discover Consumers' Sentiment," IEEE 14th International Conference on Mobile Data Management, Milan, p. 122-124.

Rosa, R. L., Rodriguez, D. Z. and Bressan, G. (2015) "Music recommendation system based on user's sentiments extracted from social networks," in *IEEE Transactions on Consumer Electronics*, vol. 61, no. 3, p. 359-367, August.

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. Stroudsburg, PA, USA: Association for Computational Linguistics, p. 417–424.

Zheng, K., Yang, Z., Zhang, K., Chatzimisios, P., Yang, K. and Xiang, W. (2016) "Big data-driven optimization for mobile networks toward 5G," in *IEEE Network*, vol. 30, no. 1, p. 44-51, January-February.

Desenvolvimento de um Objeto de Aprendizagem baseado em Mobile Learning e sistemas de recomendações para o auxílio ao processo de letramento infantil na educação básica

Saimor Raduan Araújo Souza[1], **Luis Filipe de Castro Sampaio**[1], **Lucas Felipe Alves de Araújo**[1], **Kaio Alexandre da Silva**[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO)
Caixa Postal 15.064 – 76.820-441 – Porto Velho – RO – Brazil

{saimorraduan, luiz12345, donpizza.lf}@gmail.com, kaio.silva@ifro.edu.br

Abstract. *During the process of learning a child, difficulties can be manifested in relation to reading, writing and pronouncing letters and words. To solve this problem, a Development Monitoring System is being developed whose purpose is to assist students and teachers in early childhood education through recommendations. Through an application, the system can obtain information from each student. This information can be viewed by the teacher through the Performance Monitoring System. As a result of the development of the system, it is expected to achieve a recommendation system for intervention in teaching children with learning difficulties.*

Resumo. *Durante o processo de aprendizagem de uma criança, dificuldades podem ser manifestadas em relação a leitura, escrita e pronuncia de letras e palavras. Para solucionar este problema está sendo desenvolvido um Sistema de Acompanhamento de Desenvolvimento cujo objetivo é auxiliar os alunos e professores da educação infantil através de recomendações. Por meio de um aplicativo, o sistema poderá obter informações de cada aluno. Essas informações poderão ser visualizadas pelo professor através do Sistema de Acompanhamento de Desempenho. Como resultado do desenvolvimento do sistema espera-se alcançar um sistema de recomendação para intervenção no ensino de crianças com dificuldades no aprendizado do alfabeto.*

1. Introdução

O analfabetismo, pode ser causado por vários motivos, alguns dos motivos retratados são as dificuldades em aprender a leitura, escrita e pronúncia, tanto de letras, quanto de palavras, quando agravado, isso impede que a criança possa aprender a ler e escrever, consequentemente o seu aprendizado se torna deficiente [Silva and Crenitte 2016]. Segundo os autores [Terra 2013] [Di Nucci 2003] [Soares 1998], uma pessoa é considerada analfabeta quando não possui a capacidade de adquirir informações por meio da leitura de qualquer tipo de texto, seja ele um texto escrito, processo chamando de letramento, ou uma imagem, que segundo [Stokes 2002] é o processo de letramento visual, ou seja, se após a leitura de um texto escrito ou uma imagem a pessoa não conseguir expressar as informações que nele estam contidas essa pessoa pode ser considerada um analfabeto.

Segundo [Kampff and et al 2006] qualquer material físico ou digital como, por exemplo, livro ou aplicativos mobile, que tenha sido desenvolvido para fins educacionais

e que possa ser utilizado de diferentes formas, por qualquer pessoa, pode ser classificado como um objeto de aprendizagem. Além disso, um objeto de aprendizagem pode permitir aprender em qualquer lugar. De acordo com [Kukulska-Hulme and Traxler 2005] o *Mobile Learning*, ensino a distância por meio de dispositivos móveis como, por exemplo *smartphone* e *tablet*, está cada vez se difundindo no mundo.

De acordo com [Garcia and Frozza 2013], um sistemas deve, quando requisitado, deve retornar respostas de forma rápida e com precisão a seus usuários, uma solução possível para isso é a utilização de sistemas de recomendações. Segundo [Garcia and Frozza 2013] um sistema de recomendaçoes é um sistema que utiliza informações obtidas sobre seus usuarios, rotinas e preferências, como base para poder formular uma recomendação adequada ao perfil desse usuário. Um sistema de recomnendações podem obter dados sobre os usuários de forma direta, através de formulários, ou indireta como, por exemplo, pro meio da utlização de uma aplicação.

No Brasil o analfabetismo é um problema persistente que necessita de um intervenção. Dentro desse contexto, este artigo apresenta estudos sobre uma solução que utiliza um sistema de recomedações para auxiliar professores da educação infantil em relação ao processo de letramento. O artigo apresenta na seção 2 a metodologia utilizada para o desenvolvimento do sistema e na seção 3 as conclusões e os trabalhos futuros.

2. Metodologia

Em uma sala de aula uma criança pode apresentar dificuldades em aprender o alfabeto, e as vezes por causa quantidade de alunos que um professor possuim em sua turma geralmente algumas dessas dificuldades podem não serem percebidas. Uma possivél solução para esse problema é o desenvolvimento de um Sistema de Acompanhamento baseado em recomendações, um sistema composto por: um aplicativo *Android*, um Sistema de Acompanhamento, um *Web Sevice* e um banco de dados.

As turmas que utilizarem o sistema serão divididas em dois grupos, um grupo será denominado o grupo controle e o outro o grupo experimental. O grupo controle será a metade da turma que não utilizará o aplicativo, já o grupo experimental será a metade da turma que utilizará o aplicativo. Por causa da incompatibilidade entre o banco de dados, o aplicativo e pelo sistema de acompanhamento de desenvolvimento, foi desenvolvido um *Web Service* para realizar o gerenciamento e transporte de dados entre o aplicativo e o banco de dados. Tais dados serão consumidos pelo sistema de acompanhamento de desenvolvimento através do *PHP Data Object* (PDO).

Por meio do aplicativo o sistema pode obter dados para poder criar um perfil sobre cada aluno que utilizou o aplicativo e a partir dessas informações o sistema, através do banco de dados e o *Web Service*, pode comparar o desempenho desses alunos e recomendar ao professor sobre quais letras ele deve auxiliar cada um de seus alunos.

A Figure 1 ilustra o funcionamento do objeto de aprendizagem. No sistema o *Web Service* é o responsável em realizar a comunicação entre o aplicativo e o banco de dados. O aplicativo quando é utilizado envia uma requisição para o *Web Service* que, quando é aceita, consulta o banco de dados para poder retorna uma resposta ao aplicativo, ou seja, o aplicativo consome e envia dados que são armazenados no banco de dados. O Sistema de Acompanhamento de Desenvolvimento é diretamente conectado ao banco de dados,

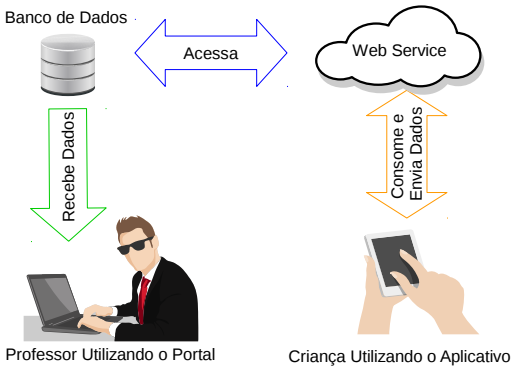

Figure 1. Fluxo de utilização do sistema

por causa dessa conexão o professor poderá visualizar os dados de cada aluno no Sistema de Acompanhamento de Desenvolvimento, ou seja, o banco de dados é consultado pelo Sistema de Acompanhamento de Desenvolvimento, que procurá os dados para poder retornar uma resposta que será exibida ao professor.

Por causa do número de pessoas no Brasil que utilizam Smarphones com o sistema *Android* optou-se em escolher essa plataforma. No aplicativo para a criação de intefaces é utilizado imagens de repositórios livres. Na home page a criança poderá escolher qualquer uma das letras do alfabeto da língua portuguesa, quando uma letra é selecionada um áudio contendo a pronúncia dessa letra é executado. Em seguida, serão exibidas imagens associadas a letra. Ao término dessas ações, uma atividade é proposta para a criança. Os dados sobre o desempenho de cada aluno, obtidas a partir do aplicativo são salvas pelo *Web Service* no banco de dados, que quando é consultado enviará esses dados para o Sistema de Acompanhamento de Desenvolvimento para serem vistos pelo professor.

O Sistema de Acompanhamento de Desenvolvimento foi desenvolvido utilizando o HTML , CSS e JavaScript, nele o professor possui a possibilidade de acompanhar o desempenho de cada um de seus aluno que utiliza o aplicativo, identificando quais as letras foram as mais e menos selecionadas, a quantidade de dias que cada aluno utilizou o aplicativo em forma de gráficos, mostrando o histórico do desempenho de cada aluno. Além disso, o Sistema de Acompanhamento de Desenvolvimento mostra as recomendação para o professor, sobre quais as letras que ele deve auxiliar a criança a aprender, com isso, o desempenho dos alunos que utilizam o aplicativo deve igualar-se com o tempo.

A Figure 2 ilustra o fluxo de utilização do aplicativo. Na home page, pagina inicial do aplicativos, todas as letras do alfabeto oficial da língua portuguesa são exibidas para que a criança possa selecionar qualquer uma para aprender, quando a letra é selecionada o *Web Service* identifica qual letra foi selecionada e salva essa informação no Banco de Dados, no perfil reverente a esse aluno, com isso, o *Web Service* manda o aplicativo reproduzir o áudio correspondente a letra selecionada. Em seguida, um interface com subdivisões, categorias de palavras, da letra selecionada é exibida, para que a criança possa selecionar. Quando a categoria é selecionada o *Web Service* salva qual a categoria foi escolhida no banco de dados e envia um comando ao aplicativo para reproduzir um áudio referente a essa categoria, ao terminar essa reprodução a criança poderá selecionar qualquer categoria.

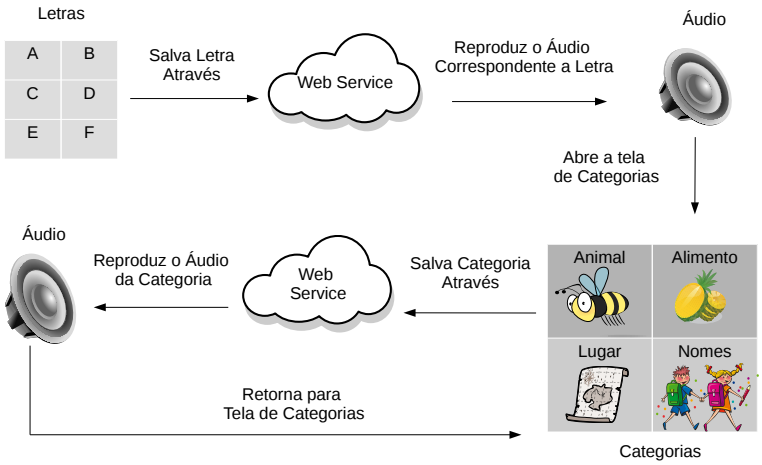

Figure 2. Fluxo de utilização do aplicativo

3. Considerações finais e trabalhos futuros

O objetivo desse sistema é provar a eficiência das técnicas utilizadas pelo sistema de recomendações na área da educação para evitar o analfabetismo. Prentende-se realizar testes em escolas que possuam o ensino fundamental, para validar a eficácia do sistema.

4. Referencias

References

Di Nucci, E. P. (2003). O letramento escolar de jovens do ensino mÃ. *Psicologia Escolar e Educacional*, 7:129 – 134.

Garcia, C. A. and Frozza, R. (2013). Sistema de recomendação de produtos utilizando mineração de dados. *Revista Tecno-Lógica*, 17:78–90.

Kampff, A. J. C. and et al (2006). Nós no mundo: objeto de aprendizagem voltado para o 1º ciclo do ensino fundamental. *RENOTE - Revista Novas Tecnologias na Educação*, 4:1–10.

Kukulska-Hulme, A. and Traxler, J., editors (2005). *Mobile learning: a handbook for educators and trainers*. Open and Flexible Learning Series. Routledge, London, UK.

Silva, N. S. M. and Crenitte, P. A. P. (2016). Desempenho de crianças com risco para dificuldade de leitura submetidas a um programa de intervenção. *CoDAS*, 28:517 – 525.

Soares, M., editor (1998). *Letramento - Um Tema Em Três Gênero*. Autêntica Editora.

Stokes, S. (2002). Visual literacy in teaching and learning: A literature perspective. *Electronic Journal for the Integration of Technology in Education*.

Terra, M. R. (2013). Letramento letramentos: uma perspectiva sócio-cultural dos usos da escrita. *DELTA: Documentação de Estudos em Linguística Teórica e Aplicada*, 29:29 – 58.

Mobility Open Data: Use Case for Curitiba and New York

Elis C. Nakonetchnei[1], Nádia P. Kozievitch[1], Cinzia Cappiello[2], Monica Vitali[2], Monika Akbar[3]

[1]Dep. de Informática, UTFPR, Curitiba, PR, Brazil

[2]Politecnico di Milano, Milan, Italy

[3]University of Texas at El Paso, El Paso, Texas, USA

elisan@alunos.utfpr.edu.br, nadiap@utfpr.edu.br,

{cinzia.cappiello,monica.vitali}@polimi.it, makbar@utep.edu

Abstract. *As cities are becoming green and smart, transportation systems are being revamped to adopt digital technologies. Open transportation data might include GIS maps for bus routes and bus stops. Such data provide a number of opportunities to analyze a segment of the current state of the transportation system of a city. In this paper, we address and discuss some of the urban mobility open data available and their characterization from the perspectives of two metropolitan cities: Curitiba (Brazil) and New York City (USA). Finally, we present challenges regarding their use.*

1. Introduction

The quality of life in a city greatly depends on the quality of its public transportation. The growing population of a city creates demand on existing infrastructure. Accommodating such demand calls for introducing new measures or changing existing infrastructure, both of which are time-consuming process and require in-depth analyses of the current state of public transportation as well as the current and future demand. One of the first steps of understanding urban transportation challenges is to understand the current state of public transportation. In this paper, we analyze the open data related to transportation for two metropolitan cities (Curitiba and New York) to gather cities open data for project EuBra-BigSea [1]. In particular, we address and discuss some of the urban mobility open data available and their characterization from the perspectives these two cities, along with challenges regarding their use.

2. Comparison

Curitiba encompasses 75 districts, and has developed and implemented mass transport corridors, densification of land-use along these corridors, and mobility solutions using Bus Rapid Transit (BRT) systems in the 1970s, where one main feature of the success of the system is its complex network of feeder lines [Duarte et al. 2016]. The city has also been participating in the open data initiative, through several government stakeholders, such as Instituto de Planejamento de Curitiba (IPPUC)[2] and the Municipality of Curitiba[3].

[1] Availableathttp://www.eubra-bigsea.eu/ - Last accessed on 11/03/2017.
[2] http://ippuc.org.br/Lastaccessedon9/03/2016
[3] http://www.curitiba.pr.gov.br/DADOSABERTOS/ – Last accessed on 09/03/2016.

General Transit Feed Specification (GTFS) data is available for buses [4], but still some differences are presented (such as missing lines, compared to the total number of bus lines available). Although Curitiba has a small network of trains, they are not used for citizen mobility (just one line, for example, is used for tourism). All the data is georreferenced and can also be downloaded at IPPUC. Curitiba does not have subways neither ferries. The datasets used in this paper come from IPPUC, the Municipality of Curitiba, along with data from Open Street Maps. In particular, the bus mobility system has the following characterization.

Bus Stops. The city has 9940 bus stops detected. Bus stops are divided among tube stations (officially 342), regular bus stops and terminals. The districts named Cidade Industrial de Curitiba (CIC) and Centro have the majority of regular bus stops, with a total of 1628 and 667 units in each one. Figure 1-D represents the bus stops in the city.

Bus Routes. The city has 482 bus routes distributed within 11 categories (*Metropolitano, Linha Direta*, among others). The categories Alimentador and Convencional have the majority of lines, with 265 and 65 units respectively. Figure 1-E represents the bus routes in the city. Different colours indicate different categories.

Bus Terminals. The city has 23 terminals (buses) and one terminal which also use trains. The oldest one is named Guadalupe, from January 1st, 1956. The districts CIC and Boqueirão concentrate their majority, with 3 and 2 units each one.

New York is composed of five boroughs: Manhattan, the Bronx, Queens, Brooklyn, and Staten Island.Metropolitan Transportation Authority (MTA) [5] runs most of the transit system of this city. There are more than 238 local routes, 62 express routes, and 7 Select Bus Service routes. GTFS data is available [6], integrating several mobility categories. The New York City Subway is the largest rapid transit system in the world by number of stations, with an average 469 stations in operation, and 25 train services. All the data is georreferenced and can also be downloaded. The datasets used in this paper come from the Department of City Planning from New York [7], the City University of New York [8], along with data from Open Street Maps. In particular, the bus mobility system has the following characterization.

Bus Routes. The city has 261 bus routes distributed within 3 categories (Express Service, Limited-Stops and Local Service). The category Local Service has the majority of lines. Figure 1-G presents the bus routes, where darker lines present intersections of routes at the same street.

Bus Stops. The city has 16,254 bus stops, and 200 bus stop names, within regular bus stops. Queens has the majority regular bus stops, with a total of 5,329. Figure 1-F presents the bus stops.

[4] `http://www.urbs.curitiba.pr.gov.br/faleconosco` – Last accessed on 02/12/2016.

[5] `http://web.mta.info/developers/developer-data-terms.html#data` – Last accessed on 09/03/2016

[6] `http://tracker.geops.ch/?z=13&s=1&x=-8226561.0174&y=4962941.5682&l=transport` – Last accessed on 02/12/2016.

[7] `http://www1.nyc.gov/site/planning/data-maps/open-data/districts-download-metadata.page` – Last accessed on 09/03/2016.

[8] `https://www.baruch.cuny.edu/confluence/display/geoportal/NYC+Mass+Transit+Spatial+Layers` – Last accessed on 09/03/2016.

Bus Terminals. The city has 3 terminals(The Port Authority Bus Terminal, George Washington Bridge Bus Station and Journal Square Transportation Center). No shapefiles were found for bus terminals.

Discussion. The distribution of the public transportation and related open data regarding bus is different along the two cities. Curitiba concentrates the bus traffic along the downtown area and some streets, but presents several bus terminals along the city. New York do not concentrate the bus lines, but present other types of transportation (such as subways and trains). Regarding data presentation, both cities have information which can be easily integrated (shapefiles along with other data types), but both also could be more helpful, for example, adding some metadata (such as date of the last update, or historical list of shapefiles) or an integrated portal, with all data sources from transportation within the municipality. Nevertheless, data from different sources (in Curitiba, for example) still present different information. Thus being necessary to integrate them to allowing a better handling of this data simultaneously.

3. Challenges and Data

In Brazil, the vehicle fleet in major cities grew more than the road structure [9]. Nevertheless, open data is still a new trend regarding data availability. Mobility challenges have already gained attention of the computer science society in Brazil [10]. In particular, these challenges can be grouped in the following areas: (i) pattern discovery, (ii) data statistics, (iii) data integration, (iv) location and tracking, (v) open and connected data, (vi) contextual information, among others. These challenges can have other subtypes such as [Kozievitch et al. 2016]: (i) **Different File Formats**: the different coordinate systems (UTM, Latitude and Longitude, etc.), and information along different sources keep the data integration as a challenge. In general, formats such as CSV, Excel, Json and Shapefile are preferred (some use APIs and KML). Not all of them provide metadata or the data visualization; (ii) **Different Reference Systems**: The data from IPPUC uses SAD69 (South American Datum from 1969), but the official standard in Brazil is SIRGAS2000 (Geocentric Reference System of the Americas). Global Positioning Systems, for example, uses WGS84 (World Geodetic System 1984). Note that in parallel, data such as street names and districts change over time. (iii) **Different File Structures / Precision / Accuracy within Data**: the official bus line data from IPPUC and URBS (listed in Figure 1-top), for example, is different. Data accuracy issue (Figure 1 - bottom left), presents the source with an atribute value which is different to its value in real world. (iv) **Open and Connected Data**: In Brazil , decrees for Open Data (decree 1135/2012, from law 12.527 [11]), competitions named Hackatons[12], as well as integration of partners for future

[9] http://www1.folha.uol.com.br/cotidiano/2014/08/1503030-frota-de-veiculos-cresce-mais-rapido-que-a-estrutura-viaria-no-pais.shtml – Last accessed on 09/03/2016.

[10] http://www.sbc.org.br/documentos-da-sbc/send/141-grandes-desafios/802-grandesdesafiosdacomputaonobrasil – Last accessed on 09/03/2016.

[11] http://multimidia.curitiba.pr.gov.br/2014/00147194.pdf – Last accessed on 09/03/2016.

[12] http://hackathon.curitiba.pr.gov.br/ – Last accessed on 09/03/2016.

projects[13] [14] are helping the data use by population. Within USA, several open data sites are available (such as Open Government [15]), but not all states are participating. (v) **Data Quality:** in order to better explore the data, metrics might be considered to verify non relevant data: errors, missing or outdated values can negatively affect decisions.

Figure 1. Different Precision/Accuracy and Values for bus data.

4. Conclusions

This paper presented an initial investigation and characterization in order to identify scenarios and implications from the urban mobility open data from the city of Curitiba and New York. Detected characteristics can be used for further analysis in order to optimize the transportation system, and contribute to standards to transportation data worldwide. This work will be extended to provide characterization in order to provide data mining and suggestions enhance mobility in both cities. **Acknowledgments**. Thanks to Municipality of Curitiba, IPPUC, NSF grant HRD-1242122 and EU-BR EUBra-BigSea project (MCTI/RNP 3rd Coordinated Call).

References

[Duarte et al. 2016] Duarte, F., Gadda, T., Luna, C. A. M., and Souza, F. T. (2016). What to expect from the future leaders of Bogotá and Curitiba in terms of public transport: Opinions and practices among university students. *Transp. R. Part F: Traffic Psychology and Behaviour*, 38:7 – 21.

[Kozievitch et al. 2016] Kozievitch, N. P., Gadda, T. M. C., Fonseca, K. V. O., Rosa, M. O., Gomes-Jr, L. C., and Akbar, M. (2016). Exploratory Analysis of Public Transportation Data in Curitiba. In *XXXVI CSBC*, pages 1656–1666. Sociedade Brasileira de Computação.

[13] `http://www.curitiba.pr.gov.br/noticias/curitiba-e-holanda-vao-trabalhar-juntas-em-projetos-de-ciclomobilidade-para-a-cidade/37601` – Last accessed on 09/03/2016.

[14] `http://multimidia.curitiba.pr.gov.br/2015/00166636.pdf` – Last accessed on 09/03/2016.

[15] `https://www.data.gov/open-gov/` Last accessed on 09/03/2016

SIRME: Sistema Inteligente de Recomendação para Matrículas Escolares

Felipe Lanzarin[1], Eder Pazinatto[1], José Maurício Carré Maciel[1]

[1]Instituto de Ciências Exatas e Geociências – Universidade de Passo Fundo – Passo Fundo – RS – Brasil

{lanzarin.felipe@gmail.com, ederpazinatto@upf.br, jmmaciel@upf.br}

Resumo. *A primeira matrícula de uma criança em fase escolar é uma atividade muito importante na vida dos pais e das crianças. A definição da escola é algo fundamental. Principalmente em grandes cidades, há várias escolas municipais de educação infantil, distribuídas geograficamente pela cidade, algumas mais próximas outras mais distantes da residência dos alunos. O SIRME é um Sistema de Recomendações que utiliza um Sistema de Informações Geográficas (SIG), para definir a melhor escola para o aluno se matricular, levando em consideração a distância e outras regras definidas pelo município ou de acordo com a necessidade dos pais.*

1. Introdução

De acordo com os funcionários da Secretaria Municipal de Educação (SME) da cidade de Passo Fundo/RS, a tarefa de distribuir os alunos do 1º ano do Ensino Fundamental nas escolas Municipais é sempre muito trabalhosa.

Quando um funcionário da prefeitura for definir a escola para um aluno, o mesmo precisa consultar todas as escolas que ainda possuem vagas e identificar a escola mais próxima do endereço do aluno olhando no *google maps*. Essa forma de trabalho é muito demorada quando o município tem um número de 35 escolas distribuídas pela cidade e em média 300 novos alunos por ano iniciando o Ensino Fundamental. Outro ponto negativo é que o funcionário deverá levar em consideração outras regras, aumentando a chance de colocar um aluno em uma escola que não seria a mais apropriada para ele.

Através do uso do Sistema de Recomendação SIRME, essa distribuição de matriculas pode ser feita com uma melhor precisão e agilidade. Nesse sistema, todas as regras são previamente definidas e utiliza também um Sistema de Informações Geográficas para identificar qual é a escola mais próxima da residência do aluno.

2. Sistemas de Recomendação

Os Sistemas de recomendação popularizaram-se ao longo dos últimos anos como mecanismos para auxiliar a escolha de filmes, músicas, notícias, livros e até mesmo pessoas com quem se relacionar. Auxiliam diretamente no aumento da capacidade de indicar de forma eficiente um conteúdo, ou uma possível decisão que será atrativa a quem está recebendo a indicação. Sendo assim, bons sistemas tem a capacidade de aumentar os resultados positivos de uma organização e por isso eles se tornam essenciais para o negócio (Reategui, Cazella 2005).

Devido a grande massa de dados que existe na internet, o utilizador se depara com o problema de não saber “por onde seguir”. A solução utilizada na maior parte dos

websites foi o advento dos motores de buscas, onde o mesmo digita uma palavra chave, e o motor de busca retorna todos os resultados que tem uma relação com essa palavra que o utilizador digitou. No entanto, os motores de procura concebidos originalmente para terem uma função utilitária, foram perdendo sua utilidade devido à existência de numerosos *conteúdos/sites* potencialmente relevantes. Por esses motivos, os sistemas de recomendação tornaram-se um tema muito atrativo e mais utilizado para recomendar conteúdos aos consumidores de informação (Ferreira; Oliveira, 2012).

2.1 Sistema de Informações Geográficas

Os Sistemas de Informações Geográficas (SIG) são sistemas e meios tecnológicos para se estudar o espaço terrestre (Pena, 2017). Existem três tipos de tecnologias que formam os SIGs: O sensoriamento remoto, o GPS e o geoprocessamento (Pena, 2017).

Sensoriamento Remoto: utiliza ferramentas como satélites e radares, para a captação de informações e imagens da superfície terrestre. São capazes de oferecer informações importantes, como a extensão de uma área, o tamanho de uma determinada cobertura vegetal, localizar focos de incêndios e desmatamentos, o movimento das massas de ar, entre outros.

Sistema de Posicionamento Global (GPS): funciona com uma cobertura de dezenas de satélite, sendo capaz de emitir informações de qualquer local do mundo, a partir das coordenadas geográficas. Pode informar posições latitude e longitude, como também endereços, traçar rotas mais curtas para se chegar a um determinado local e, até mesmo, gravar os caminhos percorridos e informar a velocidade de deslocamento.

Geoprocessamento: tratamento das informações obtidas por meio do sensoriamento remoto e do GPS para a produção de mapas, cartogramas, gráficos e sistematizações em geral. Utiliza *softwares* programados para essa função, que são capazes de adicionar legendas e informações diversas sobre o espaço representado. Uma das ferramentas de Geoprocessamento mais conhecidas e utilizadas pelas pessoas é o *Google Earth*, disponibilizado tanto em *software* quanto por meio de acesso à internet.

Por meio dos SIG é possível criar um banco de dados georeferenciados, ou seja, partindo de um referencial espacial pode se estabelecer relações espaciais e não-espaciais, permitindo que eles tenham uma localização (Lobato; Penha; Santos; Ferreira, 2008). A localização no espaço, permite recuperar e combinar informações bem como efetuar diversos tipos de análise com os dados.

3. Trabalhos Relacionados

O trabalho de Amaral e Cunha 2016, utiliza a API do Google maps Distance Matrix, para propor maneiras de medir a dificuldade imposta pela rede viária para a logistica urbana devido ao fluxo de carros, caminhões e ônibus que vem crescendo com o passar dos anos, além das restrições de fluxos de veículos pesados.

No trabalho Marini e Pazinatto 2016, foi construído um aplicativo mobile de geolocalização, para orientar a localização de prédios e salas de aulas de estudantes e visitantes dentro da Universidade de Passo Fundo.

4. Tecnologias Utilizadas

A implementação do SIG no Sistema de Recomendação foi feita com a API do Google maps Distance Matrix API[1] para calcular a distância entre dois endereços (endereço do aluno e endereço da escola). Como a API disponibiliza rotas para ônibus, carros, bicicletas e caminhadas, para este estudo a rota utilizada foi a de caminhadas.

Como a chamada da API retorna um XML, com a distância entre o endereço de origem e o de destino, foi utilizado a classe XMLReader que faz a requisição a API do google, e cria o XML em memória para posteriormente ser interpretado por uma classe que extende a classe DefaultHandler. Ambas as classes são implementadas em Java.

Para persistir o resultado e as escolas cadastradas é utilizado um framework chamado Hibernate[2] que faz a comunicação com as classes do Java e o Banco de Dados, que nesse caso é PostgreSQL[3].

5. Aplicação: SIRME

Para distribuir uma lista de alunos em escolas da rede municipal da cidade de Passo Fundo, o sistema lê um arquivo .csv que contém as informações dos alunos (definidas pela SME tais como: nome, endereço, se possui algum tipo de deficiência, se frequentava uma creche municipal, entre outras), e então faz a listagem seguindo a ordem: prioridade para alunos que tem alguma deficiência; depois para os alunos que frequentaram uma creche municipal; e ordem em que os alunos foram inscritos. O cadastro das escolas também pode ser feito por um arquivo .csv, que contém o nome da escola, endereço e a quantidade de vagas disponíveis. O sistema disponibiliza uma área de gerenciamento das escolas cadastradas conforme mostra a Figura 1.

Figura 1- Relatório das escolas cadastradas no sistema

Após ter essa lista de alunos ordenados, o sistema percorre a lista e para cada aluno calcula a distância do endereço de sua residência com o das escolas cadastradas no sistema, que ainda possuem vagas naquele momento. Depois de obter a distância da residência do aluno com cada escola, o sistema verifica qual escola tem a menor distância, e assume que o aluno será matriculado nesta escola, diminuindo o número de vagas da mesma.

[1] https://developers.google.com/maps/documentation/distance-matrix/?hl=pt-br
[2] http://hibernate.org/orm/
[3] https://www.postgresql.org

Na Tabela 1 é apresentado o resultado final, informando o nome do aluno e a escola que o sistema escolheu para ele, com a possibilidade de exportar essa tabela para um arquivo .csv, para posterior uso dos funcionários da SME. Caso o endereço de um aluno não for encontrado pela API do Google, esse endereço é apresentado na lista de erros encontrados durante a execução, conforme parte inferior da tabela 1.

	Nome	Escola	Endereço Aluno	Endereço Escola
1	Hiago Moreira	Arlindo Luiz Osório	Nagppe Kraide, 77 - 99032570, Passo Fundo	Pedro Cuimann, 385, Vila Dona Júlia, Passo Fundo
2	Lais da Silva	Antonino Xavier	Souza Neves, 75 - 99072710, Passo Fundo	Oscar Pinto, 903, Vila Jardim, Passo Fundo
3	Valentina Pedrozo	Arlindo Luiz Osório	Rua Independência, 514 - 99010041, Passo Fundo	Pedro Cuimann, 385, Vila Dona Júlia, Passo Fundo
4	Kéven da Costa	Antonino Xavier	Manoel Teixeira, 142 - 99035-040, Passo Fundo	Oscar Pinto, 903, Vila Jardim, Passo Fundo

Exportar para CSV

Erros

	Identificador	Aluno	Descrição	Endereço
1	1	Angelina Nunes	Endereço quintino bocaiuva, 65 - 99036520, Passo Fundo não encontrado!	quintino bocaiuva, 65 - 99036520, Passo Fundo

Tabela 1- Resultado da distribuição dos alunos nas escolas

6. Considerações Finais

O trabalho contribui com SME, uma vez que reduz a carga de trabalho manual que estava sendo realizada, algo que demandava maior tempo e envolvimento de várias pessoas do setor, além de facilitar o processo e também minimizar erros que podem ocorrer durante os ajustes de matrículas.

7. Referências Bibliográfica

Amaral, Julia Coutinho; Cunha, Claudio Barbieri da. Análise da Complexidade de Redes Viárias Urbanas para distribuição de Última Milha em Megacidades. Anais do XXX Congresso ANPET, Rio de Janiero/RJ. 2016. Disponível em <http://www.anpet.org.br/xxxanpet/site/anais_busca_online/documents/3_109_AC.pdf>. Acesso em 12 de março de 2017

Ferreira, Fernando C.; Oliveira, Adicinéia A. de. Os sistemas de recomendação na web como determinantes prescritivos na tomada de decisão. JISTEM - Journal of Information Systems and Technology Management. Revista de Gestão da Tecnologia e Sistemas de Informação Vol. 9, Nº. 2, pp. 353-368. 2012, São Paulo.

Lobato M. Monteiro; Penha L. Rocha da; Santos S. Baía dos; Ferreira W. Morais. A Importância dos Sistemas de Informação Geográfica (Sig's) para a Cartografia Tradicional. II Simpósio Brasileiro de Ciências Geodésicas e Tecnologias da Geoinformação. 2008, Recife.

Marini, Eliardo; Pazinatto, Eder. LocalizeUPF: Aplicativo de geolocalização no Campus I da Universidade de Passo Fundo. 1º Congresso de Simulação e Otimização do Sul – CONSOSUL. Passo Fundo/RS 2016. Disponível em <http://consosul.upf.br/images/anais2016/4_Eliardo_Marini.pdf> Acesso em 13 de março de 2017.

Pena, Rodolfo F. Alves. "SIG"; Brasil Escola. Disponível em <http://brasilescola.uol.com.br/geografia/sig.htm>. Acesso em 11 de fevereiro de 2017.

Reategui, E. Berni; Cazella, S. Cesar. Sistemas de Recomendação. XXV Congresso da Sociedade Brasileira de Computação, UNISINOS, São Leopoldo/RS. 2005

EasyTest: Plataforma Crowdsourcing para testes funcionais

Ângelo N. V. Crestani[1], Gian L. M. Flores[1], Mateus H. Dal Forno[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia Farroupilha
RS-377, Km 27 – Passo Novo – CEP 97555-000 – Alegrete – RS

{angelovieira.c,gianlucamottaflores}@gmail.com,
mateus.dalforno@iffarroupilha.edu.br

Abstract. *This paper describes the development progress of the EasyTest platform, developed from the test process proposed by DAL FORNO (2016). The platform will allow outsourcing of the functional tests execution, allowing collaboration of external testers to the organization. The use of recommendation based on profile and reputation will enable professionals evaluate the compatibility between testers profiles and test tasks, assisting in the decision making for testers recruitment in the platform.*

Resumo. *Este artigo descreve o andamento do desenvolvimento da plataforma EasyTest, desenvolvida a partir do processo de teste proposto por DAL FORNO (2016). A plataforma permitirá a terceirização da execução de testes funcionais, possibilitando a colaboração de testadores externos à organização. O uso de recomendação baseada em perfil e reputação permitirá aos profissionais avaliar a compatibilidade entre perfis de testadores e de tarefas de teste, auxiliando na tomada de decisão para o recrutamento de testadores na plataforma.*

1. Introdução

Devido a constante evolução tecnológica e a presença cada vez maior do software no cotidiano, a todo tempo surgem novas técnicas de desenvolvimento de software. Neste contexto, estratégias baseadas em inteligência coletiva vêm se destacando como alternativas as atuais metodologias de desenvolvimento de software (MAO et al., 2017).

O Crowdsourcing é uma técnica de terceirização da execução de uma pequena tarefa para uma rede indefinida de colaboradores, por meio de um convite aberto (HOWE, 2009; ZANATTA et al., 2016). Segundo Brabham (2008), o crowdsourcing inova ao remover barreiras geográficas para a execução de tarefas, ao mesmo tempo que possibilita a participação de pessoas externas à organização no desenvolvimento de novas soluções.

Para implementar o crowdsourcing são necessários três elementos (PRIKLADNICKI et al., 2014): O cliente (I), que são indivíduos que possuem uma tarefa a ser executada; A Plataforma Crowdsourcing (II) que apoia a mobilização, gerência, comunicação e remuneração pela realização das tarefas; e a Multidão (III), que são indivíduos dispersos geograficamente, responsáveis pela execução das tarefas.

O desenvolvimento de software é um dos setores que vem utilizando o crowdsourcing. Neste contexto, as principais iniciativas de plataformas se destacam para codificação e teste de software (MAO et al., 2017; ZANATTA et al., 2016).

Sistemas de recomendação são mecanismos que tem por objetivo auxiliar o usuário na obtenção de informação relevante, podendo ser por meio de seu perfil ou do

perfil de um grupo (CAZELLA; NUNES; REATEGUI, 2010). Tais perfis são criados a partir de informações obtidas de várias fontes, tais como: histórico de sites visitados, questionários, compras realizadas anteriormente, cookies do navegador, entre outros.

Com base nas informações obtidas e consolidadas em perfis, que caracterizam e qualificam os desejos de um usuário, juntamente com a avaliação de outro item semelhante, é possível recomendar produtos, notícias, comunidades, novas amizades (PAZZANI; BILLSUS, 2007). Sistemas de recomendação também podem ser utilizados para avaliação de reputação, por exemplo, a partir de feedbacks do comportamento de um determinado participante em uma comunidade (CAZELLA; NUNES; REATEGUI, 2010).

Neste contexto, o presente trabalho descreve o andamento do desenvolvimento da plataforma crowdsourcing EasyTest, cujo processo de teste é aderente ao modelo de processo de teste funcional proposto por DAL FORNO (2016), que utiliza crowdsourcing como estratégia para a execução de testes funcionais por testadores externos à organização. Pretende-se também agregar estratégias de recomendação, por meio do uso de recomendação baseada em perfis para cada um dos três elementos da plataforma crowdsourcing: Analista de teste (clientes), tarefas de teste e testadores (multidão).

2. Metodologia

O desenvolvimento do trabalho utiliza como metodologia de desenvolvimento a abordagem de entrega incremental (SOMMERVILLE, 2011), que define os requisitos e a arquitetura do sistema de maneira iterativa, e posteriormente trata o desenvolvimento, validação e integração do sistema de maneira incremental.

3. Resultados Parciais

O levantamento de requisitos da plataforma EasyTest foi desenvolvido a partir da coleta de informações do processo de teste crowdsourcing proposto por DAL FORNO (2016), bem como da realização de pesquisas a partir de outras plataformas crowdsourcing existentes. A partir dessas informações buscou-se identificar a estratégia de recomendação e de reputação adequada para uso na plataforma.

A Figura 1 apresenta a organização conceitual da plataforma EasyTest, que encontra-se em desenvolvimento. O sistema será composto por dois módulos: o módulo "Analista de teste" (responsável por elaborar e disponibilizar as tarefas de teste, bem como selecionar proposta de testador para execução da tarefa e avaliar o resultado) e o módulo "Testador" (multidão geograficamente dispersa, responsável pela execução dos testes).

Identificou-se que o modelo de recomendação adequado ao desenvolvimento da plataforma é o de modelagem de perfil baseado em conhecimento. Nesta abordagem a forma de coleta das informações para a recomendação é explícita, ou seja, o usuário é responsável por disponibilizar as informações que comporão o seu perfil através do preenchimento de um questionário (MIDDLETON; SHADBOLT; ROURE, 2004).

Em ambos os módulos (analista de teste e testador), o usuário deverá preencher seu cadastro com informações que servirão de base para a composição de seu perfil, que posteriormente será utilizado para a recomendação. A opção pela modelagem baseada em perfil se deu em função desta abordagem permitir que o usuário informe de maneira explícita suas preferências quanto às abordagens de teste, permitindo assim que as recomendações compatíveis com o seu perfil sejam exibidas.

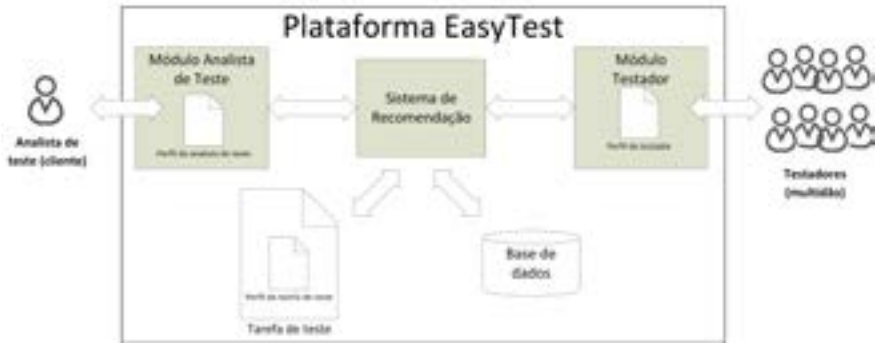

Figura 1. Organização conceitual da plataforma EasyTest.

As tarefas de teste serão cadastradas na plataforma EasyTest pelo Analista de testes. Cada tarefa será composta por uma descrição textual das atividades a serem executadas e por um perfil, que será utilizado pelo sistema de recomendação para indicar o nível de similaridade da tarefa de teste com o perfil do testador.

O testador, ao se candidatar para a execução de uma tarefa de teste, terá à sua disposição as informações que envolvem o escopo do teste a ser realizado, bem como o nível de similaridade do perfil da tarefa de teste com o seu próprio perfil. As tarefas de teste disponíveis para recrutamento serão disponibilizadas ao testador em ordem decrescente, de acordo com a similaridade da tarefa com seu perfil de teste. Adicionalmente, a plataforma permitirá que o testador personalize os critérios de exibição, possibilitando, inclusive, que tarefas com baixa similaridade com o perfil do testador sejam exibidas. O testador poderá candidatar-se para a execução de uma determinada tarefa de teste de seu interesse, propondo um valor como remuneração para sua execução.

Posteriormente, o analista de testes irá selecionar uma, dentre das propostas disponíveis para execução de uma determinada tarefa de teste. As propostas serão exibidas de acordo com o nível de similaridade entre o perfil do testador e o perfil da tarefa de teste, em ordem decrescente. Serão disponibilizadas, juntamente com cada proposta, o nível de similaridade entre o perfil da tarefa de teste e do testador, a reputação do testador na plataforma e o valor pleiteado pelo testador para a execução da tarefa. Após a conclusão da execução da tarefa de teste selecionada, o analista de testes avalia o resultado submetido pelo testador, aprovando ou reprovando a execução da tarefa e realizando o pagamento, caso o trabalho realizado seja satisfatório.

A estratégia de construção da reputação pública de cada usuário da plataforma será baseada em uma escala de estrelas (de 0 a 5), a partir da média de avaliações recebidas. Esta estratégia é utilizada, por exemplo, para avaliação de produtos em sites de comércio eletrônico. Após a conclusão de uma tarefa na plataforma, o analista de teste avaliará o trabalho realizado pelo testador, atribuindo uma nota de 0 a 5 para o profissional, juntamente com uma breve descrição textual sobre a execução da tarefa. Da mesma forma, o testador realizará o mesmo procedimento de avaliação para o analista de teste.

O desenvolvimento deste trabalho atualmente encontra-se na etapa de definição da modelagem da arquitetura do software e da base de dados. Para o desenvolvimento do sistema optou-se por uma arquitetura web baseada no padrão MVC (Model View Controller) (SOMMERVILLE, 2011), que organiza a arquitetura do sistema em camadas. Quanto à

base de dados, encontra-se em estudo qual o paradigma mais adequado a ser utilizado.

4. Considerações Finais

Neste artigo é descrito o andamento do desenvolvimento da EasyTest, plataforama crowdsourcing para a execução de testes funcionais de software. Atualmente estão em desenvolvimento a modelagem da arquitetura do sistema e a definição do paradigma da base de dados a ser utilizada.

Após a conclusão deste trabalho, o uso da plataforma EasyTest permitirá que testadores externos à organização possam contribuir por meio da execução dos testes. Um dos diferenciais na plataforma proposta está no uso de estratégias de recomendação baseadas em perfil e reputação, o que contribuirá na tomada de decisões, tanto para o analista de testes, quanto para os testadores.

Adicionalmente, pretende-se por meio da plataforma EasyTest disponibilizar uma alternativa viável às empresas de desenvolvimento de software interessadas em ampliar o uso de testes ou adequar demanda, permitindo flexibilização da mão-de-obra e redução de custos.

Referências

BRABHAM, D. C. Crowdsourcing as a model for problem solving an introduction and cases. *Convergence: the international journal of research into new media technologies*, Sage publications, v. 14, n. 1, p. 75–90, 2008.

CAZELLA, S. C.; NUNES, M. A.; REATEGUI, E. B. A ciência da opinião: Estado da arte em sistemas de recomendação. In: *JAI: Jornada de Atualização em Informática da SBC*. Rio de Janeiro: Editora da PUC Rio, 2010.

DAL FORNO, M. H. *CPFT : Uma Proposta de Processo Adaptável Para Testes Funcionais Utilizando Crowdsourcing*. Dissertação (Mestrado em Computação Aplicada) Universidade de Passo Fundo, Passo Fundo, 2016.

HOWE, J. *O Poder das Multidões*. 1. ed. Rio de Janeiro: Campus, 2009. 300 p.

MAO, K. et al. A survey of the use of crowdsourcing in software engineering. *Journal of Systems and Software*, v. 126, p. 57 – 84, 2017.

MIDDLETON, S. E.; SHADBOLT, N. R.; ROURE, D. C. D. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, v. 22, n. 1, p. 54–88, 2004.

PAZZANI, M. J.; BILLSUS, D. Content-based recommendation systems. In: ____. *The Adaptive Web: Methods and Strategies of Web Personalization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 325–341.

PRIKLADNICKI, R. et al. Brazil software crowdsourcing: a first step in a multi-year study. In: *Proceedings of the 1st International Workshop on CrowdSourcing in Software Engineering - CSI-SE 2014*. New York: ACM Press, 2014. p. 1–4.

SOMMERVILLE, I. *Engenharia de Software*. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

ZANATTA, A. L. et al. Software crowdsourcing platforms. *IEEE Software*, IEEE, v. 33, n. 6, p. 112–116, 2016.

Integração de Dados de Redutores de Velocidade no Transporte Público de Curitiba

Giovane N. M. Costa[1], Nádia P. Kozievitch[1], Keiko Fonseca[1], Tatiana Gadda[1], Rita C. G. Berardi[1]

[1]Universidade Tecnológica Federal do Paraná (UTFPR) – Curitiba, PR- Brasil

gcosta@alunos.utfpr.edu.br,{nadiap,keiko,tatianagadda, ritaberardi}@utfpr.edu.br

Abstract. *The urban transport system has increasingly provided a variety of data (sensors, bus routes, passengers, etc.). These data provide opportunities to identify traffic problems and assist the system planning and management. In this paper we address challenges and opportunities of urban data integration in a Geographic Information System (GIS) to the planning and management of speed limit enforcement devices in public transport in Curitiba - Brazil.*

Resumo. *O sistema de transporte urbano tem crescentemente fornecido uma variedade de dados (sensores, rotas de ônibus, passageiros, etc). Esses dados fornecem oportunidades para identificar problemas de tráfego e auxiliam no planejamento e gerenciamento do sistema. Neste artigo abordamos desafios e oportunidades encontrados na integração de dados urbanos em um Sistema de Informação Geográfica (SIG) para o planejamento e gestão de redutores de velocidade no transporte público em Curitiba - Brasil.*

1. Introdução

Com a crescente demanda por serviços urbanos e o volume de dados disponíveis, tem-se cada vez mais utilizado Tecnologias de Informação, como Sistemas de Informação Geográfica (SIG), para auxiliar no planejamento, gerenciamento e operação das cidades [Naphade et al. 2011].

Considerando o potencial de contribuição da sociedade e a transparência na gestão, diversos dados governamentais tem sido disponiblizados. No entanto, trabalhar com dados urbanos é considerado um dos grandes desafios da computação devido à dificuldades de integração, estatística, descoberta de padrões e gerenciamento de grande volume de dados [SBC 2015].

Neste artigo é relatada a experiência de integração de dados de radares, lombadas, linhas de ônibus e divisão de bairros de Curitiba (Brasil) em um único Sistema de Informação Geográfica. Particularmente, buscou-se identificar os desafios presentes na combinação dos diferentes dados urbanos abertos e as oportunidades do sistema integrado para planejar e gerenciar a alocação dos redutores de velocidade, assim como cumprir a legislação específica de cada um.

2. Aquisição e Integração de Dados

Em Lenzerini (2012) é apresentada uma pespectiva teórica de integração de dados, a qual é definida como "o problema de combinar dados presentes em diferentes fontes, e de prover o usuário com uma visão unificada destes dados". Em Curitiba, Barczyszyn (2015) apresenta uma primeira iniciativa de integração dos diversos dados urbanos de Curitiba com a criação de uma base geográfica unificada.

No presente trabalho, foi utilizada essa mesma base e inseridos novos dados de quatro fontes oficiais: Prefeitura Municipal de Curitiba (PMC), Secretaria Municipal de Trânsito (SETRAN), Instituto de Pesquisa e Planejamento Urbano de Curitiba (IPPUC) e da Companhia de Urbanização de Curitiba (URBS). Na Tabela 1 tem-se a estatística geral dos dados governamentais obtidos.

Tabela 1. Estatística geral dos dados governamentais obtidos.

Tema	Dados relevantes obtidos	Quantidade	Formato	Fonte
Ônibus	Trajeto das Linhas	250 linhas	JSON	PMC/URBS
Radares	Endereço e Tipo	262 radares	PDF	SETRAN
Lombadas	Quantidade por rua	1220 lombadas	PDF	SETRAN
Divisão de Bairros	Área e Limite Geográfico	75 bairros	Shapefile	IPPUC

Os radares foram geocodificados pelo *Google Street View*[1]. Além do uso dos dados governamentais, foi realizado também um estudo de caso in loco na Linha Interbairros II Anti-horário para a obtenção de outro conjunto de dados, com o registro manual de todas as lombadas e radares do trajeto por fotos georreferenciadas.

Figura 1. Diferentes tipos de redutores: lombada física (A), faixa de travessia elevada (B), controlador eletrônico (C) e lombada eletrônica (D).

Na Figura 1 é exemplificado os diferentes dispositivos registrados. Para a análise, os dispositivos representados em (A) e (B) foram considerados lombadas. Já os dispositivos representado em (C) e (D) foram considerados radares.

3. Desafios de Integração e Oportunidades

Apesar da iniciativa e avanço da abertura de dados governamentais pelos gestores públicos, diversos desafios foram encontrados para sua integração em Sistemas de Informação Geográfica, como:

- A lista de lombadas não pode ser utilizada por mencionar apenas a quantidade de lombadas por ruas e nenhuma outra referência geográfica.

1 https://www.google.com/streetview/, acesso em 27 jan. 2017.

- A localização de cada radar é dado pelo endereço do imóvel mais próximo e pelas ruas que o interseccionam (Ex: Sete de Setembro, 1610 – entre Ubaldino do Amaral e José de Alencar). Além da imprecisão desse método quando comparado com o de coordenadas geográficas e a necessidade de geocodificação para trabalhar em SIG, foi particularmente trabalhoso encontrar os radares em algumas rodovias e grandes avenidas, em que não havia nenhuma casa ou estabelecimento numerado ao redor.
- O trajeto de cada linha de ônibus é definido por um conjunto de pontos sem parâmetros de ordenação e com vários pontos repetidos. Para formar o desenho do trajeto (Figura 2 B) foi necessário encontrar visualmente um subconjunto ordenado no software QGIS.

Ressalta-se também os diferentes formatos para os arquivos encontrados e o uso de formatos não estruturados (como PDF), que são práticas já conhecidas como não adequadas para dados abertos por dificultar a edição e inserção direta em banco de dados.

Na Figura 2 A apresenta-se a distribuição de radares por bairros em Curitiba. Cores escuras indicam um número maior de radares. Já na Figura 2 B é apresentado o resultado da coleta in loco, junto ao trajeto da linha de ônibus e ao mapa da cidade pelo *OpenStreetMap*[2].

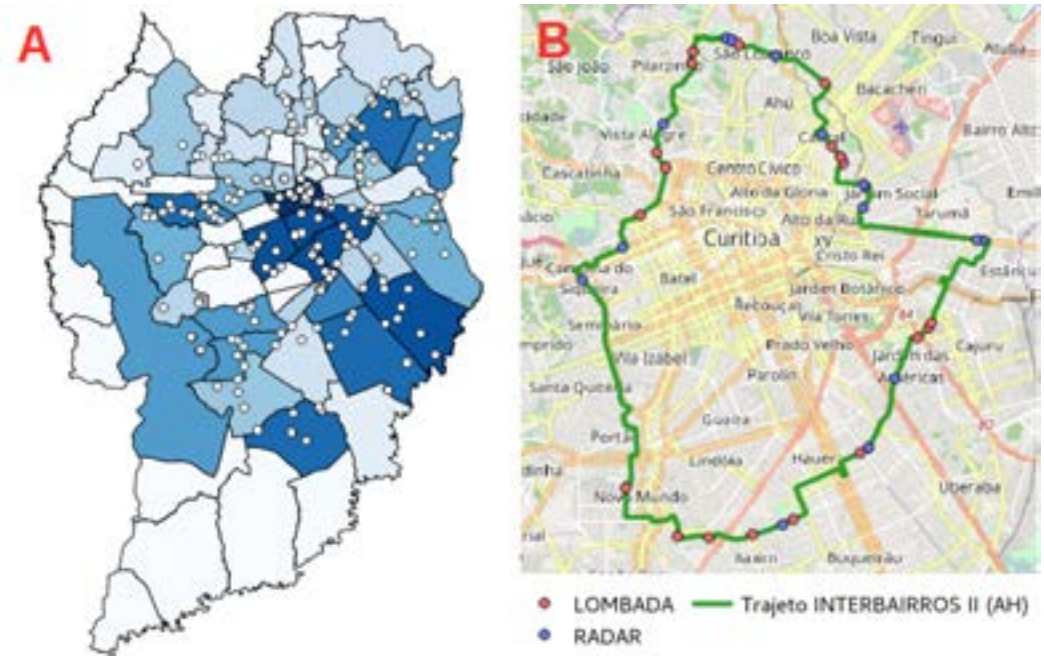

Figura 2. Distribuição de radares por bairro em Curitiba (A); e distribuição de radares e lombadas na linha Interbairros II (B).

Nota-se na Figura 2 A que a região central é a mais fiscalizada. O bairro Centro (o mais escuro na figura) possui a maior quantidade de radares (26), seguido por seis bairros ao sul com menos da metade (10 radares). Já no estudo de caso in loco (Figura 2 B) foram registrados quinze radares e dezenove lombadas.

2 https://www.openstreetmap.org/, acesso em 27 jan. 2017.

A legislação de lombadas[3] e radares[4] dispõem que antes e depois da instalação dos mesmos, sejam feitos estudos técnicos envolvendo o histórico de acidentes para avaliar a necessidade e eficiência do dispositivo no local. Para lombadas prevê-se ainda restrições para instalação das mesmas em curvas e vias em que circulam linhas regulares de transporte coletivo, assim como uma distância mínima entre elas.

Com a inclusão da data de instalação dos equipamentos nos dados existentes e uma integração com dados de acidentes, a utilização de um SIG como o aqui desenvolvido pode facilitar os estudos técnicos e permitir que essas condições legais citadas acima sejam verificadas.

4. Conclusão

Este trabalho apresentou os desafios encontrados na integração de dados abertos de redutores de velocidade e oportunidades para o planejamento urbano. Como resultado, destaca-se a falta de dados governamentais georreferenciados, completos e estruturados.

Esse resultado mostra que além da abertura dos dados governamentais, para utilização em sistemas de planejamento e gestão do transporte urbano, há oportunidades de melhoras na coleta de dados, com padronização e georreferencimento por sistemas de coordenadas, e também na disponibilização dos dados, com o uso preferencial de formatos *machine-readable*.

Como trabalho futuro, considera-se a integração de dados históricos de acidentes, assim como a análise dos dados de percurso dos ônibus. Essa integração pode permitir investigar o impacto dos redutores nos acidentes e na mobilidade urbana.

5. Agradecimentos

Agradecemos à Prefeitura Municipal de Curitiba, ao IPPUC, ao SETRAN, à CAPES, ao CNPq e ao projeto EU-BR *EUBra-BigSea* (*MCTI/RNP 3rd Coordinated Call*).

Referências

Barczyszyn, G. L. (2015). Integração de dados geográficos para planejamento urbano da cidade de Curitiba. Trabalho de Conclusão de Curso, Departamento de Informática, UTFPR.

Lenzerini, M. (2002). Data integration: A theoretical perspective. InProceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS '02, pages 233–246, New York, NY, USA. ACM.

Naphade, M., Banavar, G., Harrison, C., Paraszczak, J., and Morris, R. (2011). Smarter cities and their innovation challenges. Computer, 44(6):32–39.

SBC (2015). Grandes desafios da computação no Brasil. Sociedade Brasileira de Computacão. Disponível em <http://www.sbc.org.br/documentos-da-sbc/send/141-grandes-desafios/802-grandesdesafiosdacomputaonobrasil>, acesso em 11 jan. 2017.

3 http://www.denatran.gov.br/images/Resolucoes/Resolucao6002016_new, acesso em 27 jan. 2017.

4 http://www.denatran.gov.br/download/Resolucoes/RESOLUCAO_CONTRAN_396_11, acesso em 7 jan. 2017.

Uma Ferramenta Online para Execução de Scripts em SQL

Marcos V. de Moura Lima[1], Paulo R. Rodegheri[1], Jean Luca Bez[2], Neilor A. Tonin[1]

[1]Departamento de Engenharias e Ciência da Computação
Universidade Regional Integrada do Alto Uruguai e das Missões (URI)
Erechim – RS – Brasil

[2]URI Online Judge – Erechim – RS - Brasil

{marcos,jean,neilor}@urionlinejudge.com.br, prr@uricer.edu.br

Abstract. *This paper presents an online tool for running SQL scripts, integrated with the URI Online Judge website. The URI SQL is a tool in final phase of development, which will assist students and teachers in the database disciplines, presenting a visual environment for submitting and testing scripts in query language.*

Resumo. *Este artigo apresenta uma ferramenta online de execução de scripts SQL, integrada com a ferramenta URI Online Judge. O URI SQL é uma ferramenta em fase final de desenvolvimento que apoiará alunos e professores nas disciplinas de Banco de Dados, apresentando um ambiente visual para submissão e testes de execução de scripts em linguagem de consulta.*

1. Introdução

No método tradicional de ensino de Banco de Dados, especialmente no que diz respeito à construção de *scripts* SQL, o professor geralmente passa uma lista de exercícios sobre algum comando de manipulação de dados, em seguida os alunos devem então executar seus comandos em SQL sobre uma base previamente montada e "populada". Logo após, o aluno compara a sua resposta do exercício com a resposta do professor.

Analisando esse método tradicional de ensino descrito acima, alguns problemas podem ser detectados como, por exemplo. Em turmas grandes o acompanhamento de cada aluno pelo professor pode vir a ser algo muito trabalhoso e pouco produtivo, tendo em vista que cada aluno tem uma forma de entender o conteúdo e dificuldades individuais. Para (Sadiq and Orlowska, 2004), as atividades práticas individuais são muito importantes, pois o fato de o SQL ser uma linguagem não procedural, requer que o estudante aprenda a pensar dentro da lógica de conjuntos, em vez de algoritmos.

Compreendendo as dificuldades no ensino das linguagens de consulta, o URI SQL integrado com o portal de programação URI Online Judge, está sendo desenvolvido para atuar como uma ferramenta de apoio para os professores e alunos das disciplinas de Banco de Dados. O objetivo geral da ferramenta é proporcionar ao aluno um ambiente de execução e verificação online de *scripts* SQL. Essa verificação é feita de forma automática pelo *judge* da ferramenta.

Para a melhor compreensão do contexto relacionado ao URI-SQL, a sessão 2 apresenta o ambiente onde o módulo está inserido: o portal URI Online Judge. As funcionalidades, ambiente de submissão e formas de uso estão descritas na sessão 3. Por fim, a sessão 4 apresenta os trabalhos relacionados e a sessão 5 as conclusões.

2. URI Online Judge

O portal URI Online Judge (*www.urionlinejudge.com.br*) é uma ferramenta que auxilia alunos e professores nas disciplinas da Ciência da Computação. O projeto vem sendo desenvolvido na URI – Universidade Regional Integrada – Campus de Erechim, desde 2011. A ferramenta conta com problemas no estilo ICPC (*International Collegiate Programming Contest*) da ACM. Além disso, os usuários podem testar suas soluções com juízes online. O projeto foi apresentado publicamente pela primeira vez nos Estados Unidos, no WorldComp'12 [Tonin and Bez 2012].

3. URI Online Judge SQL

O URI Online Judge SQL está sendo desenvolvido para ser uma ferramenta visual para submeter e testar a execução de *scripts* SQL. O objetivo da ferramenta é criar um ambiente de prática diferenciado para o aluno resolver seus exercícios em uma plataforma agradável e interativa.

Nesta nova ferramenta os problemas são divididos em quatro assuntos: seleção de dados, inserção de dados, atualização de dados e criação de tabelas. Além disso, como no URI Online Judge, o URI-SQL conta com uma divisão de níveis nos exercícios, assim qualquer aluno sendo iniciante ou avançado poderá usar a ferramenta. Esses níveis são determinados inicialmente pelos autores dos problemas, mas também são influenciados pela sugestão dos usuários que já resolveram os exercícios.

Figura 1. Um problema em linguagem SQL.

A Figura 1 apresenta o problema **Select Básico**, exemplificando como os problemas em SQL estão estruturados. Para auxiliar aos alunos, todo problema tem uma descrição contendo as informações necessárias que permitem chegar à resposta correta. Além da descrição, os exercícios têm um exemplo do esquema do Banco de Dados e uma tabela com os dados de saída esperados para a entrada que foi apresentada.

3.1. Ambiente de Submissão

Inicialmente as submissões recebidas são executadas pelo SGBD Postgres, na versão 9.3. O sistema do URI SQL é genérico, ou seja, permite facilmente a inclusão de novos Bancos de Dados relacionais como o MySQL e Oracle.

Para cada problema do URI SQL, existem *scripts* que geram e preparam o ambiente de submissão para o usuário. Esses *scripts* têm a função de criar as tabelas e inserir os dados iniciais nas mesmas. Eles são ativados quando uma submissão é escalonada para ser julgada. A Figura 2 ilustra uma solução para o **Select Básico**.

SOURCE CODE

```
1  select name from customers where state = 'RS';
2
```

Figura 2. Exemplo de código para submissão do problema Select Básico.

Se a solução do usuário for executada sem erros no SGBD, os dados retornados pela consulta do usuário são comparados com um arquivo resposta contendo os dados esperados para a solução do exercício em questão.

Algumas dúvidas podem ser levantadas sobre essa forma de comparação, como por exemplo: dois *scripts* diferentes podem retornar os mesmos dados, mas são consultas com propósitos diferentes. Porém a chance de ocorrer isso é pequena, pois quando um exercício é criado, atenta-se para esse fato, tentando evitar que estas situações se apresentem. Além disso o aluno não terá acesso aos *scripts* que rodam diretamente no SGBD. No entanto ele poderá baixar os *scripts* de exemplo e testar em um ambiente local, fazendo com que ele tenha um maior controle e *feedback* da sua solução.

Figura 3. Resultado da submissão do problema "Select Básico".

Ao final da avaliação da submissão, um *script* é executado para limpar e resetar o ambiente, deixando o mesmo pronto para avaliar uma submissão de outro usuário. É então retornado um código representando a resposta que o usuário irá visualizar através da interface web da ferramenta, como mostra a Figura 3 acima.

As possíveis respostas são: *Accepted* – a solução para o exercício está correta; *Runtime Error* – o *script* do usuário não foi executado no SGBD (geralmente por erros de sintaxe); *Presentation Error* – Essa resposta ocorre quando os dados retornados têm erros na quantidade de espaços em branco ou inversão de letras maiúscula por letras minúsculas. Este erro pode ocorrer em consultas mais elaboradas, quando envolve tratamento dos dados; *Wrong Answer* – a solução para determinado problema está errada; *Time Limit Exceeded* – a solução do usuário estourou o tempo limite definido para cada problema.

4. Trabalhos Relacionados

O SQLator é uma ferramenta web interativa para aprender SQL [S. Sadiq, 2004]. A plataforma conta com exercícios em *SELECT* e interface na língua inglesa. O URI-SQL se difere da ferramenta comparada pelos seguintes fatores: O aluno tem a possibilidade de resolver os problemas em 4 assuntos diferentes (*SELECT, UPDATE, INSERT e CREATE*), a interface web é moderna e simples e os problemas da ferramenta tanto como a interface web estão em 3 diferentes linguagens (Inglês, Português e Espanhol).

5. Conclusões

Sabendo dos problemas que o método tradicional de ensino a linguagens de consulta possui, o URI Online Judge SQL, será uma ferramenta de apoio aos estudantes e aos professores das disciplinas de Banco de Dados. A ferramenta proporcionará aos alunos a possibilidade de testar várias vezes e quando quiserem suas soluções para os problemas, auxiliando no aprendizado das linguagens de consulta SQL. Com relação aos professores, eles não precisarão se preocupar com a criação dos ambientes, ou seja, com a preparação das tabelas, dos dados e com a correção da solução do aluno. Desta forma sobra ao professor mais tempo para acompanhar os alunos nas suas dificuldades.

Integrando a ferramenta ao portal de programação URI Online Judge, espera-se que o alcance da ferramenta seja muito grande, tendo em vista o impacto que o portal tem hoje, tanto no Brasil como no resto do mundo.

Referências

S. Sadiq, M. Orlowska, W. Sadiq, and J. Lin. Sqlator: an online sql learning workbench. In Proceedings of the 9th Annual SIGCSE Conference on innovation and Technology in Computer Science Education, 2004.

Shazia Sadiq and Maria Orlowska. SQLator: An Online SQL Learning Workbench. In in Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education ITiCSE ’04, page 30, 2004.

The ACM International Collegiate Programming Contest (ICPC). Disponível em: http://icpc.baylor.edu/. Acesso: 06 de fevereiro de 2017.Knuth, D. E. (1984), The TeXbook, Addison Wesley, 15th edition.

Tonin, N. A. and Bez, J. L. (2012). URI Online Judge: A New Classroom Tool for Interactive Learning. In WORLDCOMP’12 – The 2012 World Congress in Computer Science, Computer Engineering, and Applied Computing, volume 1, pages 242-246

Palestras Convidadas

Sistemas de Recomendação: o que, quando, onde, como você quer, e nem sabia!

Mirella M. Moro

Sobre a autora: Mirella M. Moro é professora adjunta do Departamento de Ciência da Computação (DCC) da Universidade Federal de Minas Gerais (UFMG). Possui doutorado em Ciência da Computação pela University of California in Riverside (2007), e graduação e mestrado em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (UFRGS). É membro do Education Council da ACM (Association for Computing Machinery). Foi Diretora de Educação da SBC (Sociedade Brasileira de Computação, 2009-2015), editora-chefe da revista eletrônica SBC Horizontes (2008-2012), e editora associada do JIDM (Journal of Information and Data Management, 2010-2012). Seus interesses de pesquisa estão na área de Banco de Dados, incluindo tópicos como processamento de consultas, disseminação e recomendação de conteúdo, redes sociais e bibliometria.

Como a Nuvem e o Big Data estão moldando as empresas do século 21 e onde estão as oportunidades para os profissionais de TI

Fabio Elias

Sobre o autor: Fabio Elias é um executivo de Desenvolvimento de Negócios de Produtos Oracle, com 25 anos de experiência em Tecnologia da Informação em consultoria estratégica, aplicações empresariais e tecnologias em nuvem. Fabio Elias entrou na Oracle há 19 anos tendo passado por diferentes linhas de negócios e conta hoje com uma visão abrangente do mapa de portfólio da nuvem Oracle, liderando iniciativas para ajudar os clientes corporativos a obterem benefícios através do uso dessas soluções. Fabio Elias possui MBA em Gestão de Tecnologia da Informação pela FIA (Fundação Instituto de Administração) de São Paulo e é graduado em Tecnologia de Processamento de Dados através do Centro de Ensino Superior de Maringá-PR.

"O que você quer ser quando... se formar?"

Carina Friedrich Dorneles

Sobre a autora: Atua no Departamento de Informática e Estatística da UFSC em pesquisa, ensino, administração e orientação de estudantes nos níveis de IC, graduação, mestrado e doutorado. Concluiu seu doutorado em Ciência da Computação em 2006, e mestrado em Ciência da Computação em 2000, pela Universidade Federal do Rio Grande do Sul (UFRGS). Durante o período de doutorado, realizou estágio sanduíche na University of Washington, Seattle, EUA, no grupo de pesquisa de Banco de Dados e Inteligência Artificial. Seus interesses de pesquisa incluem as áreas de Gerenciamento de Dados, Recuperação de Informação, Mineração de Dados com ênfase na Web, Descoberta de Conhecimento e Extração e Matching de Informação. Participa de projetos de colaboração internacional, dentre eles o projeto VIDAS, com a França, dentro do programa CAPES/COFECUB. Em 2005, foi co-idealizadora da Escola Regional de Banco de Dados, e da Sessão de Demos do Simpósio de Banco de Dados. Atuou como Editora da Coluna Bits, Bytes e Batom da revista eletrônica SBC Horizontes.

Grandes Desafios da Pesquisa em Computação no Brasil

Renata Galante

Sobre a autora: Renata Galante tem mestrado e doutorado em Ciência da Computação pela Universidade Federal do Rio Grande do Sul. É Professora associada associada no Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS). Atualmente é chefe do Departamento de Informática Aplicada, do Instituto de Informática da UFRGS. É também Diretora Administrativa da Sociedade Brasileira de Computação (SBC). Está envolvida com atividades de extensão, ensino de graduação e pós-graduação e orientação de alunos de mestrado e doutorado. Tem trabalhado em diversos projetos de pesquisa financiados por agências de fomento como CNPq, Finep e Fapergs. Desenvolve pesquisa nas áreas de Banco de Dados, Big Data, Redes Sociais, Cidades Inteligentes.

Deep Learning

Rodrigo Coelho Barros

Sobre o autor: Rodrigo Coelho Barros é bacharel em Ciência da Computação pela Universidade Federal de Pelotas (2007), mestre em Ciência da Computação pela PUCRS (2009) e doutor em Ciências de Computação e Matemática Computacional pela Universidade de São Paulo (2013). Recebeu os prêmios da Sociedade Brasileira de Computação (SBC) e da CAPES de melhor tese em Ciência da Computação do país (2014). Atualmente é professor adjunto da Faculdade de Informática da PUCRS, onde atua tanto na graduação quanto na pós-graduação, sendo um dos líderes do Grupo de Pesquisa em Inteligência de Negócio e Aprendizado de Máquina (GPIN). É Bolsista de Produtividade do CNPq (nível 2), e coordena projeto de grande porte em parceria com a empresa Motorola desde 2015. Seus principais interesses de pesquisa são o aprendizado de máquina e a mineração de dados, com foco atual em redes neurais profundas.

Minicursos

Sistemas de Recomendação em Repositórios Digitais.

Roberto Willrich

Resumo: Repositórios Digitais (RDs) oferecem funcionalidades para gerenciar, armazenar e acessar conteúdos digitais de diversos tipos, como teses, dissertações, artigos científicos, imagens, áudios e vídeos. Uma característica comum a qualquer solução de RD é o uso de metadados para descrever os seus conteúdos. Com sua popularização, é crescente o número de conteúdos disponibilizados nos RDs, gerando o problema clássico da sobrecarga de informação. A solução considerada neste minicurso para tratar este problema são os sistemas de recomendação. Este minicurso tem como objetivo apresentar os principais conceitos e técnicas utilizadas em sistemas de recomendação para RDs. Mais especificamente, neste minicurso abordaremos: a) padrões e soluções de RDs e metadados, b) motivação do uso de sistemas de recomendação em RDs, c) técnicas de recomendação aplicadas a RDs, d) implantação de sistemas de recomendação em soluções abertas de RDs através de Web Services e Ontologias, e) ilustração através de estudos de caso de sistemas de recomendação baseado em Web Service para Rds.

Sobre o autor: É Professor Titular do Departamento de Informática e Estatística (INE) da Universidade Federal de Santa Catarina (UFSC). Possui doutorado em Informática pela Université Paul Sabatier - França (1996), graduação e mestrado em Eng. Elétrica pela UFSC (1987 e 1991). Ele realizou estágio Pós-doutoral no Laboratoire d´Analyse et d ´Architecture des Systèmes (LAAS-CNRS) no período de 2005-2006. Tem experiência na área de Multimídia, Qualidade de Serviços, Repositórios Digitais, Sistemas de Recomendação, Anotações Digitais e Web Semântica.

Sistemas de Recomendação e Dados Interligados

Giseli Rabello Lopes

Resumo: O problema da sobrecarga de informação continua sendo experienciado pelos usuários ao tentarem acessar informações nos mais variados meios. Nesse contexto, surgiu a necessidade do desenvolvimento de aplicações capazes de recomendar, de maneira personalizada, itens que possam ser de interesse de um usuário em particular. Para satisfazer essa necessidade, Sistemas de Recomendação vêm sendo concebidos para os mais variados domínios de conhecimento, salientando-se seu amplo uso em sites de comércio eletrônico. Evidencia-se também a evolução da Web, nos últimos anos, de um repositório distribuído de documentos conectados através de hiperlinks, para um repositório distribuído de dados interligados em formato RDF (*Resource Description Framework*). Essa evolução abriu oportunidades para explorar o potencial poder do uso de dados interligados no enriquecimento da descrição dos itens a serem recomendados. Além disso, Sistemas de Recomendação vêm sendo desenvolvidos para auxiliar no próprio processo de publicação de dados interligados. Este minicurso tem por objetivo apresentar os principais conceitos e técnicas utilizadas por estas duas áreas de pesquisa: "Sistemas de Recomendação" e "Dados Interligados", e os avanços que vêm sendo obtidos por utilizá-las em conjunto.

Sobre a autora: Atua como Professora Adjunta do Departamento de Ciência da Computação (DCC) da Universidade Federal do Rio de Janeiro (UFRJ). Possui doutorado em Ciência da Computação pela Universidade Federal do Rio Grande do Sul - UFRGS (2012), mestrado em Ciência da Computação pela UFRGS (2007) e graduação em Engenharia de Computação pela Fundação Universidade Federal do Rio Grande - FURG (2004). Já atuou como professora substituta junto ao Departamento de Informática Aplicada da UFRGS e como pós-doutoranda na Pontifícia Universidade Católica do Rio de Janeiro - PUC-Rio junto ao Departamento de Informática - DI. Tem experiência na área de Ciência da Computação, com ênfase em Sistemas de Informação e Banco de Dados, atuando principalmente nos seguintes temas: Sistemas de Recomendação, Dados Interligados, Recuperação de Informação, Redes Sociais e Web Semântica.

Introdução à Recuperação de Informações

Viviane Moreira

Resumo: A Recuperação de Informação (RI) trata do armazenamento, indexação e busca por informações de natureza não estruturada (texto, imagem, vídeo, etc.). Interagimos diariamente com sistemas de RI, seja utilizando motores de busca na Web, ou procurando por e-mails em nossos computadores. Esse minicurso tem por objetivo apresentar os principais conceitos da área de RI para dados textuais. Serão abordadas as etapas de pré-processamento, indexação, consulta e coleta de dados na Web.

Sobre a autora: Professora do Instituto de Informática da UFRGS, onde desempenha atividades de pesquisa e de ensino tanto na graduação como na pós-graduação. É bolsista de produtividade em pesquisa do CNPq (nível 2). Completou doutorado em Ciência da Computação na Middlesex University em Londres (2004) e mestrado em Ciência da Computação na UFRGS (1999). Suas áreas de pesquisa são Bancos de Dados, Recuperação de Informações e Mineração de Textos. A professora ministra a disciplina de Recuperação de Informações no PPGC da UFRGS há dez anos, tem orientado trabalhos de pesquisa e redigido artigos científicos nesta área.

Sistemas de Recomendação e suas Aplicações

Sílvio César Cazella

Resumo: A busca pelo conteúdo/item que melhor atenda o perfil de um usuário, de modo a satisfazer a sua necessidade tornou-se um desafio em um cenário onde as opções podem ser inúmeras. Quando pensamos em sistemas que podem a partir de um banco de dados oferecer uma quantidade considerável de opções ao usuário (exemplos como, plataformas que oferecem filmes, ou plataformas que oferecem Objetos de Aprendizagem), a busca manual pelo usuário por este conteúdo/item pode ser exaustiva e resultar em conteúdos/itens não totalmente satisfatório uma vez que o usuário não possui uma visão do todo. Neste contexto de demanda e grande oferta de conteúdo/itens surgem os Sistemas de Recomendação como uma possível solução na identificação do que poderia ser de maior interesse ou não para um usuário alvo, ou grupo de usuários. Este minicurso tem como objetivo apresentar os conceitos, técnicas, estratégias, limitações dos Sistemas de Recomendação, buscando apresentar aplicações acadêmicas e comerciais. Algoritmos clássicos e soluções híbridas serão apresentados, bem como tendências para área apresentadas no ACM Recommender Systems conference (RecSys).

Sobre o autor: Concluiu o doutorado em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (2006), tendo realizado doutorado "sanduiche" na Universidade de Alberta no Canadá (2003-2004). Mestre em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (1997). Graduado em Informática pela Pontifícia Universidade Católica do Rio Grande do Sul (1993). Atualmente é Professor Adjunto - Nível IV no departamento de Ciências Exatas e Sociais Aplicadas da Universidade Federal de Ciências da Saúde de Porto Alegre. Professor efetivo do Programa de Pós-Graduação em Ensino na Saúde (PPGENSAU/UFCSPA), colaborador dos Programas de Pós-Graduação em Ciência da Saúde (PPGCS/UFCSPA) e Programas de Pós-Graduação em Informática na Educação (PPGIE/UFRGS). Possui uma série de trabalhos publicados sobre Sistemas de Recomendação. Orientou uma série de trabalhos na área de Sistemas de Recomendação e participa de projetos focados na Aplicação de Sistemas de Recomendação em Educação.

Oficinas

Usando Apache Mahout em Sistemas de Recomendação

Daniel Lichtnow, Joedeson Fontana Junior

Resumo: Pesquisas na área de Sistemas de Recomendação ganharam força nos anos 90, período no qual foram desenvolvidos uma série de métodos e algoritmos. Muitos destes estão consolidados, sendo hoje incorporados a muitas aplicações. Alguns destes algoritmos foram reunidos em *frameworks* de forma a facilitar sua utilização por desenvolvedores e pesquisadores. Dentre os *frameworks* disponíveis está o *Apache Mahout*, que possui alguns algoritmos usados em Sistemas de Recomendação, estando dentre estes os relacionados à abordagem referenciada como Filtragem Colaborativa. Nesta oficina, serão inicialmente descritos os algoritmos usados em Sistemas de Recomendação com ênfase nos algoritmos relacionados a Filtragem Colaborativa. Após, será demonstrada sua utilização usando o *Apache Mahout*. Serão ainda mostrados pequenos exemplos de como usar os algoritmos oferecidos pelo *Apache Mahout* dentro de aplicações, usando pequenos exemplos em Java.

Sobre os autores:

Daniel Lichtnow: Professor adjunto da Universidade Federal de Santa Maria (UFSM) onde atua no Colégio Politécnico na graduação em Sistemas para Internet e no Técnico em Informática. Possui doutorado em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (2012), mestrado em Ciência da Computação pela Universidade Federal de Santa Catarina (2001) e graduação em Tecnologia em Processamento de Dados pela Universidade Católica de Pelotas (1990). Tem interesse pelos seguintes temas: Bancos de Dados, Sistemas de Recomendação, Recuperação de Informações e Qualidade de Dados/Informações na Internet.

Joedeson Fontana Junior: Aluno do 6º semestre de graduação do curso de Sistemas para Internet da Universidade Federal de Santa Maria (UFSM). Trabalha desde 2015 como bolsista de iniciação científica nas áreas de Sistemas de Recomendação, Computação Ubíqua e Pervasiva e Banco de Dados Geográficos.

Otimização de Consultas MySQL

Daniel Di Domênico

Resumo: O desempenho de um sistema e de um banco de dados dependem basicamente de boas práticas. Esta oficina foca em técnicas de otimização de consultas que ajudam a otimizar o tempo de resposta do banco de dados para sistemas de informação, descrevendo algumas das principais técnicas associadas à escrita de queries otimizadas.

Sobre o autor: Bacharel em Ciência da Computação pela Universidade de Passo Fundo. Especialista em Sistemas Web Universidade de Passo Fundo. Diretor e Sócio da TN3 Soluções com sólida experiência na área de Tecnologia da Informação e da Comunicação, atuando no desenvolvimento de diversos projetos no Brasil e Exterior. Pela TN3 atuou na coordenação do projeto de NFe no grupo Coca-Cola, Aunde Brasil, Grendene e demais multinacionais. Atuou como professor universitário nos cursos de Sistemas de Informação e Design Web na Universidade Luterana do Brasil (ULBRA) até o ano de 2008 e na escola de Sistemas de Informação da Faculdade Meridional (IMED) até o ano de 2010. Vice-Presidente do PoloSul.org gestão 2016/2017.

Projeto de Banco de Dados Relacional com a Ferramenta brModeloWeb

Ronaldo dos Santos Mello

Resumo: esta oficina destina-se à teoria e à prática de projeto de Bancos de Dados Relacionais (BDRs). O foco da mesma são as etapas de modelagem conceitual e modelagem lógica de BDR. Dicas avançadas de modelagem entidade-relacionamento (ER) são apresentadas, bem como regras detalhadas de mapeamento ER-relacional. A oficina é incrementada com exercícios práticos de projeto de BDR através da utilização de uma nova versão da ferramenta acadêmica brModelo denominada brModeloWeb. A ferramenta é uma aplicação Web e permite desde a modelagem conceitual até a geração do script SQL para a criação do BDR.

Sobre o autor: Ronaldo dos Santos Mello possui graduação, mestrado e doutorado em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (UFRGS). Atualmente é professor asssociado do Departamento de Informática e Estatística da Universidade Federal de Santa Catarina (INE/UFSC), onde atua nos cursos de graduação e como docente permanente no Programa de Pós-graduação em Ciência da Computação (PPGCC/UFSC). Tem experiência na área de Banco de Dados (BD), atuando principalmente nos seguintes temas: modelagem de dados, integração e interoperabilidade de dados e BDs para Big Data. Coordena, desde 2006, o Grupo de pesquisa em BD da UFSC (GBD/UFSC). Coordenou, de novembro de 2011 a novembro de 2015, o PPGCC/UFSC. Atua como revisor de diversas conferências e periódicos nacionais e internacionais e orienta trabalhos em nível de graduação, mestrado e doutorado. Possui mais de setenta trabalhos publicados e já ministrou diversas palestras, minicursos e oficinas na área de BD. É um dos criadores da brModelo, uma das ferramentas mais utilizadas pela comunidade acadêmica brasileira para projeto de BDs relacionais.

Oficina Para Meninas

Nádia Puchalski Kozievitch, Silvia Amélia Bim

Resumo: Esta oficina tem como objetivo instigar 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 historicamente 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 crianças1 2, tutoriais de SQL 3, 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 as autoras:

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.

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.

Banco de Dados Distribuídos em PostgreSQL - Replicação na Prática

Álvaro Nunes Melo

Resumo: A oficina irá abordar os conceitos de arquitetura do PostgreSQL que formam a base para todas as formas de replicação nativas, em um modelo de hands-on. Da mesma forma, serão realizados pilotos utilizando os diversos tipos de replicação disponíveis, e suas aplicações práticas. Log shipping, warm e hot standby, bem como o uso de slots físicos de replicação serão alguns dos temas abordados.

Sobre o autor: Diretor da Atua Sistemas de Informação e DBA PostgreSQL há mais de quinze anos. Já foi professor em Instituições de Ensino Superior, e hoje contribui com estas difundindo o uso do PostgreSQL em disciplinas de Banco de Dados, bem como com palestras em eventos como o Fórum Internacional de Software Livre, PgDays e semanas acadêmicas.

Gerando Recomendações Usando Filtragem Colaborativa e RecDB

Gláucio Ricardo Vivian

Resumo: Os Sistemas de Recomendação estão visivelmente presentes no nosso cotidiano, o que muitas vezes não percebemos é todos os conceitos envolvidos e o conjunto de tecnologias necessárias para o seu funcionamento. No campo dos Sistemas de Informação, a recomendação é uma área de pesquisa atual e que se desenvolveu muito nos últimos anos devido ao Netflix Prize e uma grande demanda por conteúdo personalizado na Web. As suas principais aplicações são no comércio eletrônico, redes sociais, conteúdos sob demanda, gastronomia, música, pesquisa científica, dentre outros. Duas áreas que estão intimamente relacionadas são os Sistemas Recomendação e Banco de Dados, o projeto RecDB (*Fork* do PostgreSQL) apresenta uma solução unificada para este fim. 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 a recomendação baseada em filtragem colaborativa. Mais especificamente abordaremos: a) a motivação para a área, os diferentes tipos de sistemas de recomendação, e suas aplicações, b) a fundamentação teórica e os principais conceitos; c) filtragem colaborativa do tipo item-usuário e item-item; d) fatoração de matrizes (SVD); e) gerando recomendações com SQL em banco de dados relacionais utilizando o RecDB; f) avaliando as recomendações; e g) datasets disponíveis para experimentos (Netflix, Movielens, entre outros).

Sobre o autor: Atua como Analista de Tecnologia da Informação no Instituto Federal Farroupilha - Campus Frederico Westphalen. Possui ampla experiência na indústria de software nas áreas de Sistemas de Informação, Banco de Dados e Desenvolvimento Java. Tem interesse em pesquisas nas áreas de Sistema de Recomendação, Big Data, Data Science, Smart Cities, Data Mining, Recuperação de Informação e Cientometria. É incentivador de projetos de software open source e autor principal dos projetos MahoutGUI e Xml2Arff. Possui graduação em Ciência da Computação (2009), especialização em Desenvolvimento Web com Java (2013), é mestrando do Programa de Pós-Graduação em Computação Aplicada (PPGCA) da UPF.

ERBDPASSO FUNDO2017

www.sbc.org.br/erbd2017

Realização

SBC Sociedade Brasileira de Computação

UPF Universidade de Passo Fundo | PPGCA Programa de Pós-Graduação em Computação Aplicada Instituto de Ciências Exatas e Geociências - ICEG

Patrocínio Institucional

FAPERGS

TechDEC TECNOLOGIA EM MOVIMENTO

Apoio

PoloSul.org

APLTEC

Patrocínio Temático

Compasso

ORACLE

Patrocínio Promocional

atua

Renove