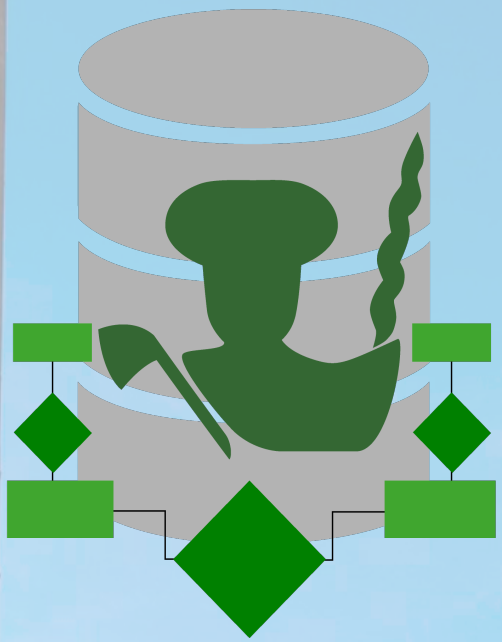


ANAIS

XV ERBD

Escola Regional de Banco de Dados

2019

Chapecó - SC

INTELIGÊNCIA DE DADOS

Organização:

UNIVERSIDADE FEDERAL DA FRONTEIRA SUL

UNOESC

Realização:

SBC Sociedade Brasileira de Computação

Patrocínio:

CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico

CRESOL Crédito para seus sonhos

amplimed Gestão de Clínicas

Apoio:

NTIC Núcleo das Empresas de Tecnologia da Informação e Comunicação

deatec Associação Polo Tecnológico do Oeste Catarinense

www.sbc.org.br/erbd2019

/erbd.sbc

XV ESCOLA REGIONAL DE BANCO DE DADOS

10 a 12 de abril de 2019
Chapecó – SC – Brazil

ANAIS

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

Organização
Universidade Federal da Fronteira Sul - UFFS Chapecó
Universidade do Oeste de Santa Catarina - Unoesc Chapecó

ISSN: 2177-4226

E77a Escola Regional de Banco de Dados (15. : 2019 : Chapecó, SC)
Anais [da] XV Escola Regional de Banco de Dados / XV Escola Regional de Banco de Dados, 10 a 12 abril 2019, Chapecó, SC ; promoção: Sociedade Brasileira de Computação, organização: Universidade Federal da Fronteira Sul, UFFS, Universidade do Oeste de Santa Catarina, Unoesc. – Chapecó : UFFS, 2019. – 119 p. : il.
Tema: Inteligência de dados.
ISSN: 2177-4226

1. Computação. 2. Eventos – Computação. 3. Banco de dados. I.Título. II. Inteligência de dados. III. Universidade Federal da Fronteira Sul. IV. Universidade do Oeste de Santa Catarina.

CDD 004

Ficha catalográfica elaborada pela
Divisão de Bibliotecas – UFFS
Nelcy T. da Rosa Kegler
CRB – 14/1311

Editorial

A Escola Regional de Banco de Dados (ERBD), evento anual da região sul do Brasil promovido pela SBC, chega a sua 15a edição em 2019, com o tema Inteligência de Negócios. Criada com o objetivo de promover e divulgar a área de banco de dados e suas áreas afins, ao longo desses anos a ERBD vem oportunizando a divulgação e a discussão de trabalhos, além da integração e troca de ideias entre os participantes. A programação do evento inclui sessões técnicas, oficinas, minicursos e palestras proferidas por profissionais e pesquisadores de destaque na comunidade brasileira. A cada edição é escolhido um tema, a partir de um assunto de destaque na área.

Desde sua criação, a ERBD tem sido realizada em diferentes cidades dos estados do Rio Grande do Sul, Santa Catarina e Paraná. Em 2019 ocorreu de 10 a 12 de abril, organizada pela Universidade Federal da Fronteira Sul (UFFS) e pela Universidade do Oeste de Santa Catarina (UNOESC), ambas do Campus Chapecó – SC, na cidade de Chapecó - SC. Agradecemos ao Comitê de Organização Local da ERBD, coordenado pelos Profs. Guilherme Dal Bianco (UFFS) e Denio Duarte (UFFS), que trabalharam arduamente para garantir a organização e o bom andamento do evento.

As sessões técnicas oferecem espaço para a apresentação de trabalhos submetidos em duas categorias: Pesquisa e Aplicações/Experiências. Todos os artigos submetidos foram avaliados por pelo menos 3 membros do Comitê de Programa. A categoria de Pesquisa recebeu 11 submissões, das quais 8 artigos foram aceitos e apresentados em períodos de 20 minutos. A categoria de Aplicações/Experiências recebeu 11 submissões, das quais 8 artigos foram aceitos e apresentados em períodos de 10 minutos, bem como na forma de pôster.

Os Anais da XV ERBD são o resultado do esforço coletivo de um grande número de pessoas entusiasmadas e dedicadas. Gostaríamos de agradecer aos membros do Comitê de Programa que fizeram revisões de excelente qualidade e nos prazos estipulados. Finalmente, agradecemos aos autores que submeteram seus trabalhos para a ERBD e a todos os participantes que vieram prestigiar a sua realização.

Helena Graziotin Ribeiro, UCS
Presidente do Comitê de Programa da ERBD 2019

Carta do Coordenador Geral

Com muita satisfação que realizamos pela primeira vez no oeste de Santa Catarina a Escola Regional de Banco de Dados – ERBD nas dependências da Universidade Federal da Fronteira Sul- UFFS e da Universidade do Oeste de Santa Catarina – UNOESC.

O evento, que encontra-se na sua XV edição, contou com a presença de cerca de 200 participantes, incluindo estudantes de ensino técnico, de graduação e de pós-graduação, bem como profissionais da academia e da indústria de TI da região da grande Oeste de Santa Catarina. O tema Inteligência de Dados foi selecionado devido a demanda atual de sistemas e processos para extração de conhecimento a partir de bases de dados, norteando as palestras, minicursos, oficinas, painéis e sessões técnicas.

Ao todo, 9 palestrantes e 16 artigos colaboraram com suas experiências acadêmicas e profissionais, proporcionando um evento de alta qualidade, contribuindo para a atualização e qualificação do público que prestigiou o evento.

O grupo de professores, técnicos administrativos e estudantes envolvidos na organização foi fundamental para que a ERBD 2019 fosse executada com sucesso. Muito obrigado pela disponibilidade e responsabilidade na execução das mais diversas tarefas, de forma voluntária e colaborativa. Agradeço ainda ao Comitê Diretivo da ERBD, à Comissão Especial de Banco de Dados e aos demais colaboradores da Sociedade Brasileira de Computação. Por fim, meu agradecimento especial à Universidade Federal da Fronteira Sul e à Universidade do Oeste de Santa Catarina, por terem cedido estrutura física e recursos humanos para todo o suporte necessário à realização do evento.

Guilherme Dal Bianco, UFFS
Coordenador Geral da ERBD 2019

XV Escola Regional de Banco de Dados

10, 11 e 12 de Abril de 2019
Chapecó – SC – Brazil

Apoio

Sociedade Brasileira de Computação – SBC

Organização

Universidade Federal da Fronteira Sul – UFFS – Chapecó
Universidade do Oeste de Santa Catarina – Unoesc – Chapecó

Comitê Diretivo

Daniel dos Santos Kaster – UEL (Presidente)
Daniel Luis Notari – UCS
Eduardo Nunes Borges – FURG

Coordenações

Comitê de Programa: Helena Graziotin Ribeiro – UCS

Palestras: Karin Becker – UFRGS

Minicursos: Luiz Celso Gomes Jr. – UTFPR

Oficinas: Solange Pertile – UFSM

Comitê Organizador Local

Adriano Sanick Padilha – UFFS
Andressa Sebben – UFFS
Centro Acadêmico Computação – UFFS
Denio Duarte – UFFS
Fernando Bevilacqua – UFFS
Guilherme Dal Bianco – UFFS, Coordenador Geral
Graziela Simone Tonin – UFFS

Jean Assmann Ferro – UFFS
Raquel Aparecida Pegoraro – UFFS
Tiago Daniel de Braga – UFFS
Tiago Zonta – Unoesc

Comitê de Programa

Alcides Calsavara – PUC-PR
Ana Marilza Pernas – UFPel
André Schwerz – UTFPR
Angelo Frozza – IFC - Camboriú
Carina F. Dorneles – UFSC
Carmem Hara – UFPR
Cristiano Cervi – UPF
Daniel Kaster – UEL
Daniel Lichtnow – UFSM
Daniel Notari – UCS
Deborah Carvalho – PUC-PR
Deise Saccol – UFSM
Denio Duarte – UFFS
Edimar Manica – IFRS
Eduardo Borges – FURG
Flávio Uber – UEM
Geomar Schreiner – UFSC
Guilherme Dal Bianco – UFFS
Gustavo Kantorski – UFRGS
Helena Ribeiro – UCS
João Marynowski – UFPR
Karina S. Machado – FURG
Luiz Celso Gomes Jr – UTFPR
Marcos Aurelio Carrero – UFPR
Nádia Kozievitch – UTFPR
Raquel Stasiu – UTFPR / PUC-PR
Raqueline Penteado – UEM
Rebeca Schroeder – Udesc
Regis Schuch – UNICRUZ
Renata Galante – UFRGS
Renato Fileto – UFSC
Ronaldo Mello – UFSC
Sandro Camargo – Unipampa
Scheila de Avila e Silva – UCS
Sergio Mergen – UFSM
Solange de L. Pertile – UFSM

Sumário

Artigos Completos de Pesquisa

Completos

Especificando um Middleware para a Interoperabilidade do Registro Eletrônico em Saúde

Carlos Andrew Costa Bezerra[1], André Magno Costa de Araújo[2]

[1]Departamento de Sistemas de Informação
UNITPAC – Araguaína, TO – Brazil

[2]Departamento de Sistemas de Informação
Universidade Federal de Alagoas (UFAL) – Penedo, AL – Brazil

andrew@r2asistemas.com.br, andre.araujo@penedo.ufal.br

Abstract. *This work specifies an HL7-based middleware capable of encoding, storing and interoperating Electronic Health Record data. Based on the HL7 clinical document architecture, the software architecture of the proposed middleware was specified, a set of rules were described to map the information of a relational data schema in HL7 messages and a tool was implemented to support the data interoperability through the proposed solution.*

Resumo. *Este trabalho especifica um middleware em nuvem baseado no padrão HL7 capaz de codificar, armazenar e interoperar os dados do Registro Eletrônico em Saúde (RES) entre organizações do setor de saúde. Baseado na arquitetura de documentos clínico do HL7, especificou-se a arquitetura de software do middleware proposto, descreveu-se um conjunto de regras que mapeiam as informações de um esquema de dados relacional em mensagens HL7 e implementou-se uma ferramenta que dá suporte a interoperabilidade do RES por meio da solução proposta.*

1. Introdução

Os Sistemas de Informação em Saúde (SIS) processam diariamente uma larga quantidade de informações que auxiliam as organizações em saúde em suas atividades operacionais e administrativas. Desde que o uso do papel foi minimizado para registrar as informações do Registro Eletrônico em Saúde (RES), muito se tem discutido sobre o uso de padrões, normas e procedimentos no desenvolvimento de SIS. Conforme determinam as boas práticas de órgãos internacionais [IEEE 2008], os SIS devem prover mecanismos de segurança e unicidade do RES, preservando o histórico e a evolução dos dados clínicos, podendo este ser reutilizado e compartilhado por outros domínios da área da saúde.

Em um domínio da saúde, é comum o uso de diferentes aplicações para gerenciar áreas/departamentos que lidam diretamente com os cuidados do paciente, como a anatomia patológica, diagnóstico por imagem, análises clínicas e Prontuário Eletrônico do Paciente (PEP). Nesse sentido, a heterogeneidade dos tipos de dados, a falta de padrão para uniformizar os atributos de dados e as terminologias do RES e as diferentes tecnologias utilizadas para desenvolver SIS, dificultam o processo de troca de dados entre as organizações de saúde (e.g., hospitais, operadoras de saúde e órgãos governamentais).

Atualmente os padrões ISO/EM 13606 [ISO 2008], HL7 [Noumeir and Pambrun 2010] e openEHR [Beale and Heard 2007] representam importantes iniciativas que auxiliam e melhoram o ciclo de desenvolvimento de aplicações em saúde. Enquanto os padrões ISO/EM 13606 e openEHR tratam de questões sobre como armazenar e uniformizar os atributos de dados e as terminologias do RES, o padrão HL7 fornece um conjunto de especificações que visam padronizar a troca e o transporte de informações entre SIS. Diversas pesquisas desenvolvidas pela indústria de software e a academia apontam o padrão HL7 com uma alternativa viável para se alcançar a interoperabilidade entre aplicações de saúde [Bezerra et al. 2015]. Nesse sentido, algumas soluções baseadas em HL7 foram desenvolvidas para facilitar a troca de dados entre organizações privadas e públicas e interoperar dados de aplicações heterogêneas dentro de uma mesma organização. Além disso, grandes empresas da área de Tecnologia da Informação (TI) como IBM e Siemens investem em soluções de interoperabilidade de dados baseadas no padrão HL7 [IBM 2016], [Siemens 2016].

Embora o padrão HL7 venha sendo debatido e utilizado nas mais diversas áreas da saúde, percebe-se a falta soluções de software baseadas no padrão HL7 que permitam o mapeamento dos dados de um SIS legado e façam a interoperabilidade do RES com outras organizações de saúde. Nesse sentido, este trabalho especifica um middleware baseado no padrão HL7 capaz de interoperar os dados do RES por meio de um serviço em nuvem que mapeia, codifica, persiste e sincroniza os dados entre SIS. Para isso, especificou-se a arquitetura de software do middleware proposto, descreveu-se um conjunto de regras que mapeiam as informações de um esquema de dados relacional em mensagens HL7 e implementou-se uma ferramenta que dá suporte à interoperabilidade do RES por meio da solução proposta.

As demais seções deste artigo estão organizadas da seguinte forma. A seção 2 contextualiza os conceitos básicos utilizados no desenvolvimento deste e trabalho e traz uma análise dos principais trabalhos correlatos identificados no estado da arte. A seção 3 apresenta e discute a solução proposta, enquanto a seção 4 exemplifica a interoperabilidade de dados utilizando o middleware desenvolvido. Por fim, a seção 5 descreve as considerações finais deste artigo.

2. Conceitos Básicos e Trabalhos Correlatos

Esta seção descreve os conceitos básicos utilizados para o desenvolvimento deste trabalho (Seção 2.1) e comenta as principais contribuições dos trabalhos correlatos identificados no estado da arte (Seção 2.2).

2.1. Conceitos Básicos

Esta seção está organizada da seguinte forma. A seção 2.1.1 aborda a proposta do padrão HL7 para a interoperabilidade de dados no setor de saúde, enquanto a seção 2.1.2 contextualiza os fundamentos de middleware.

2.1.1. HL7

Health Level - 7 (HL7) é um padrão internacional que contém um conjunto de normas para a transferência de dados clínicos e administrativos entre aplicativos de software usados por organizações da área da saúde. Esse padrão é baseado na camada 7 do modelo Open System for Intercomunication (OSI) [ISO 1996]. O padrão HL7 engloba grupos de

especificação como mensagem de protocolos para a troca de informações entre sistemas de saúde e Arquitetura de Documento Clínico (CDA) para troca de documentos. O padrão também oferece eventos como gatilhos para disparar as mensagens para os sistemas de software que estão interligados por ele. Esses gatilhos são eventos de contexto real como a internação de um paciente. Quando ocorre uma internação em uma aplicação de software, uma mensagem no formato HL7 será construída com as informações do paciente e do tipo da internação. Essa mensagem será encaminhada para todos os outros sistemas de software que necessitam interoperar os dados.

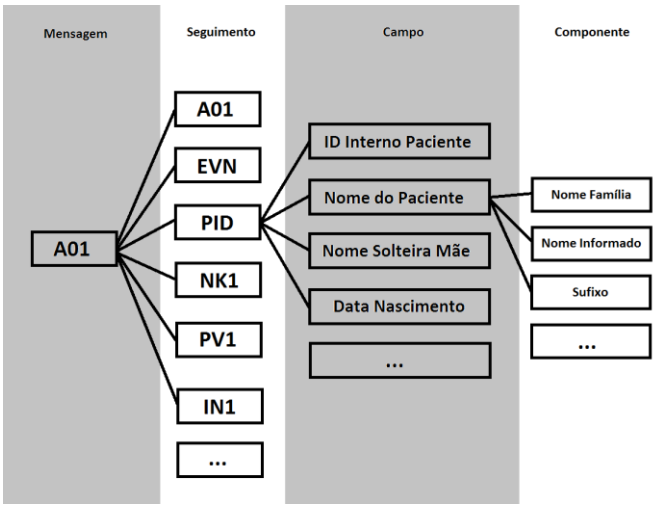


Figura 1. Estrutura de mensagem HL7.[Petry et al. 2005]

Os segmentos da mensagem são divididos em campos que respeitam uma ordem pré-determinada e possuem um tipo de dado cada. Um campo pode ser subdividido por sua vez em componentes que devem ser utilizados em cada caso. A Figura 1 ilustra a estrutura de uma mensagem no padrão HL7.

2.1.2. Middleware

Middleware é uma tecnologia para aplicações distribuídas capaz de tornar transparente os detalhes de rede e lidar com uma grande quantidade de funcionalidades de alto valor para o desenvolvimento, a implantação, a execução e a interação de aplicações [Ibrahim 2009]. A ideia principal é ser como um intermediário entre duas camadas proporcionando a comunicação entre as partes conectadas. Não se trata somente de uma aplicação de rede para conexão entre dois lados, mas tem por objetivo promover a interoperabilidade entre as aplicações, protegendo detalhes da implementação de funcionalidades e fornecendo um conjunto de interfaces para colaboração entre os clientes [Liu et al. 2012].

Existem tipos de middlewares que podem ser implementados com o objetivo de realizar a troca e a interoperabilidade de dados, sendo esses tipos: transacionais, procedurais, orientados a mensagens e orientados a objetos.

O middleware proposto neste trabalho é caracterizado como sendo orientado a mensagens. Os elementos essenciais para um middleware orientado a mensagens são: os clientes, as mensagens e o provedor que inclui uma interface de programação de

aplicações e ferramentas para administração da troca de mensagens entre os clientes. A forma da troca de dados do middleware orientado a mensagens é assíncrona.

2.2. Trabalhos Correlatos

Investigando os trabalhos correlatos sobre middlewares voltados para a interoperabilidade de dados utilizando o padrão HL7, identificou-se as seguintes pesquisas.

A solução proposta por [Liu et al. 2012] consiste em um middleware extensível baseado em HL7 para prover um canal de comunicação entre diferentes sistemas de informações em saúde que não suportavam a troca de mensagens HL7. [Ko et al. 2006] desenvolveram uma solução orientada a arquitetura (SOA) que oferece um serviço de troca de mensagens HL7 por meio de WEB Services. Mitre hData é um framework de troca de dados eletrônicos de saúde baseado na WEB com interfaces compatíveis com o padrão Fast Healthcare Interoperability Resources (FHIR) [MITRE Corporation 2015]. Mirth Connect é um middleware de código aberto projetado para troca de mensagens no padrão HL7; e conta com ferramentas para desenvolvimento, teste, implantação e monitoramento de interfaces [Meta Healthcare 2015].

Os trabalhos acima citados representam um importante avanço no estado arte, no entanto, percebe-se que eles não oferecem recursos para que as organizações em saúde façam a codificação de mensagens HL7 utilizando o mapeamento de dados diretamente de um esquema de dados. O middleware proposto neste trabalho tem como principal característica permitir o mapeamento dos dados de um SIS legado e interoperar o RES com outras organizações de saúde.

3. Middleware para Interoperabilidade de Dados Baseado no HL7

Esta seção apresenta e descreve o middleware proposto para interoperar os dados do RES entre organizações de saúde.

3.1. Arquitetura e Visão Geral

A arquitetura de software de um sistema define os seus componentes, suas propriedades externas e de seus relacionamentos com outros sistemas de software. A arquitetura desenvolvida para o serviço proposto consiste em um conjunto de componentes de software que interagem entre si (Figura 2).

O middleware desenvolvido neste trabalho é composto por dois componentes principais chamados de hCloud Middleware e hCloud Client. hCloud Middleware se refere ao módulo que fica hospedado em uma arquitetura computacional em nuvem, enquanto o hCloud Client se refere a uma aplicação local que consome os serviços do componente em nuvem.

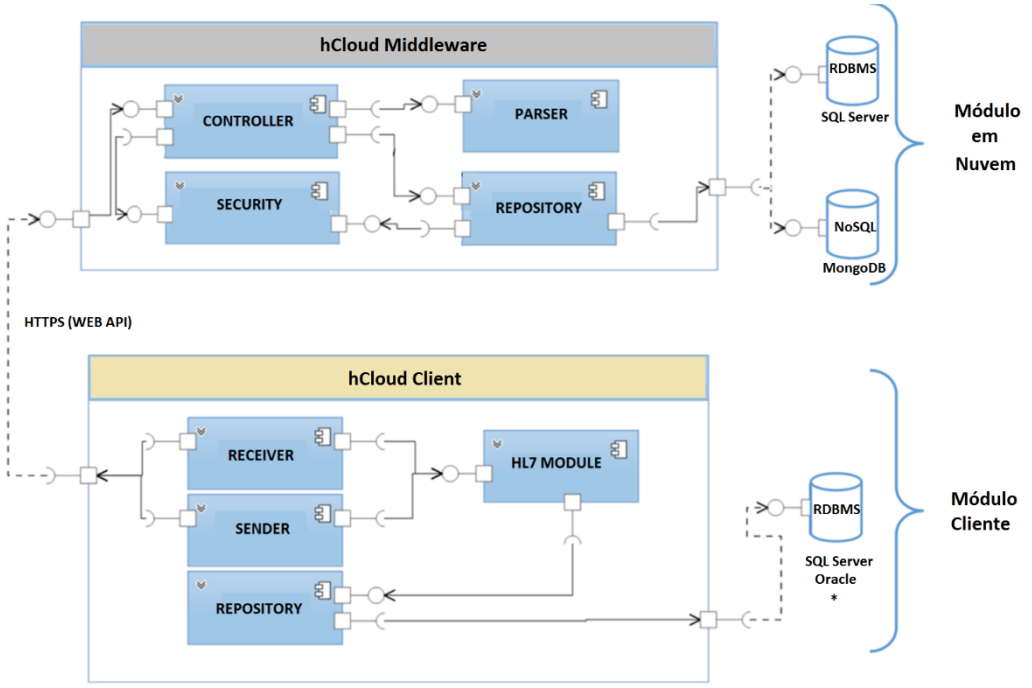


Figura 2. Arquitetura do hCloud Middleware e hCloud Client

Para o desenvolvimento de hCloud Middleware e hCloud Client, fez-se uso de um padrão de projeto que tem como principal característica separar as regras de negócios da aplicação, da camada de persistência de dados. O intuito dessa abordagem é diminuir a dependência da tecnologia de banco de dados em relação a lógica da aplicação. Para que isto ocorra, utiliza-se interfaces para mediar a comunicação entre os componentes de software da arquitetura. Assim, hCloudClient poderá se conectar a diferentes sistemas de bancos de dados (e.g., Oracle, dBase, SQL Server) e isolar os detalhes técnicos de cada tecnologia em um componente, ocultando os detalhes e possibilitando que cada nova tecnologia de banco de dados seja acoplável ao projeto sem grandes esforços de programação. A seguir, detalha-se os componentes principais da arquitetura ilustrada na Figura 2.

3.2. hCloud Client

O hCloud Client é responsável por fornecer as funcionalidades necessárias para que uma instituição de saúde possa construir mensagens HL7 a partir das informações armazenadas em seu esquema de dados. Para que ocorra a troca de mensagens no padrão HL7, o componente hCloud Client conecta e sincroniza os dados com o componente em nuvem (i.e., hCloud Middleware). Além disso, o hCloud Client conta com outros subcomponentes que são detalhados a seguir.

O componente Receiver é responsável por contatar o hCloud Middleware e, havendo mensagens disponíveis para o cliente em questão, recebe e encaminha as mensagens para o componente chamado HL7 Module. O componente HL7 Module especifica os formatos das mensagens e verifica se as mesmas estão de acordo com as especificações do padrão HL7.

O componente Repository tem duas funções básicas: gerenciar a persistência dos dados que são recebidos por meio do Receiver e manter os métodos que representam as

regras de negócio para o hCloud Client. Essas regras envolvem a seleção de informações, a configuração de gatilhos no banco de dados e o mapeamento que é feito entre os campos do padrão HL7 e os campos do esquema de dados relacional da instituição de saúde.

Figura 3. Atividade de configuração.

Conforme ilustra o diagrama da Figura 3, uma vez que a configuração estiver concluída, o hCloud Client entrará em operação e receberá qualquer evento disparado pelos gatilhos configurados no banco de dados. Além disso, codificará as informações relacionadas com o evento disparado para uma mensagem HL7 utilizando as especificações do HL7 Module.

Para o processo de mapeamento das informações do esquema de dados relacional para o padrão HL7, as seguintes regras são consideradas: i) toda mensagem deve estar relacionada a um evento de saúde (e.g., admissão do paciente); ii) toda mensagem deve seguir o layout de composição de mensagem especificado pelo padrão HL7; iii) toda mensagem será construída assim que o gatilho que representa o evento de saúde for acionado no banco de dados; iv) toda mensagem deve conter segmentos de informações compostas por campos das tabelas do banco de dados.

3.3. hCloud Middleware

O hCloud Middleware é um serviço que é executado em uma infraestrutura de nuvem computacional e tem a responsabilidade de receber as mensagens enviadas pelas instituições de saúde que utilizam o serviço de interoperabilidade do hCloud Client. Essa estrutura em nuvem possibilita que todas as instituições de saúde tenham um único ponto de compartilhamento e que diminuam a quantidade de configurações de infraestrutura de rede computacional.

O middleware atende por chamadas assíncronas e conta com a implementação de interfaces que facilitam a manutenção e atualização dos seus componentes. O hCloud Middleware conta com os seguintes subcomponentes: Controller, Repository, Parser e Security. A definição de cada um deles é dada a seguir.

O Controller é responsável por oferecer serviços em nuvem ao hCloud Client. Existem dois tipos de serviços oferecidos pelo Controller: i) solicitação de sincronização e ii) envio de mensagens HL7. A solicitação de sincronização ocorre quando as instituições de saúde, por meio do hCloud Client, solicitam mensagens HL7 advindas de outras instituições que estão disponíveis no banco de dados em nuvem. Já o envio de mensagens ocorre quando o hCloud Client envia uma mensagem HL7 para ser compartilhada por meio do hCloud Middleware para as outras instituições de saúde que utilizam esse serviço em nuvem.

Quando uma operação de sincronização é solicitada, o Controller aciona o módulo responsável por autenticar e autorizar o uso dos serviços oferecidos pelo hCloud Middleware (i.e., Módulo Security). O módulo Security utiliza um par de chaves particulares para garantir a confidencialidade, a integridade, e a autenticidade das informações trocadas entre as instituições de saúde. Esse par de chaves é utilizado para criptografar a informação trafegada entre o hCloud Client e o hCloud Middleware.

O Repository tem a função de manipular os dados que chegam e que saem do hCloud Middleware. Além disso, contém as regras de negócio para a troca de mensagens entre as instituições de saúde, como o método para fornecer uma lista de mensagens que não foram enviadas para outras instituições e a funcionalidade para evitar redundância de mensagens. Por fim, o módulo Parser verifica se a mensagem recebida está em conformidade com a especificação do padrão HL7.

4. Exemplificando a Interoperabilidade de Dados por Meio do Middleware

As informações persistidas nas instituições de saúde estão estruturadas com base em tecnologias e plataformas diferentes. Em virtude da heterogeneidade existente no armazenamento de dados (i.e., diferentes sistemas de gerenciamento de banco de dados), existe uma dificuldade considerável para se interoperar os dados entre as instituições de saúde.

Uma das principais características do serviço de interoperabilidade é permitir que os dados do RES sejam codificados em mensagens HL7 a partir de um banco de dados legado. Nesse sentido, foi desenvolvida uma interface amigável e de fácil configuração para tornar viável o serviço de interoperar os dados entre aplicações de saúde.

Após a especificação da arquitetura do middleware, desenvolveu-se uma ferramenta que permite o mapeamento dos dados de um esquema relacional para o formato de mensagem HL7. Para isso, o hCloud Client conta com uma funcionalidade de mapeamento para cada grupo de eventos especificados no padrão HL7.

A funcionalidade de mapeamento ilustrada na Figura 4, contém um conjunto de requisitos de dados do padrão HL7 (e.g., informações do paciente) que, para cada evento selecionado, é possível escolher o campo/coluna de uma tabela existente no esquema de dados relacional de uma aplicação de saúde. Além disso, é possível configurar gatilhos que transmitem as mensagens codificadas do serviço cliente para a nuvem. Toda vez que um paciente for admitido em uma instituição de saúde, o hCloud Client dispara os dados em forma de mensagem HL7 para ser persistida em um sistema de banco de dados NoSQL do hCloud Middleware.

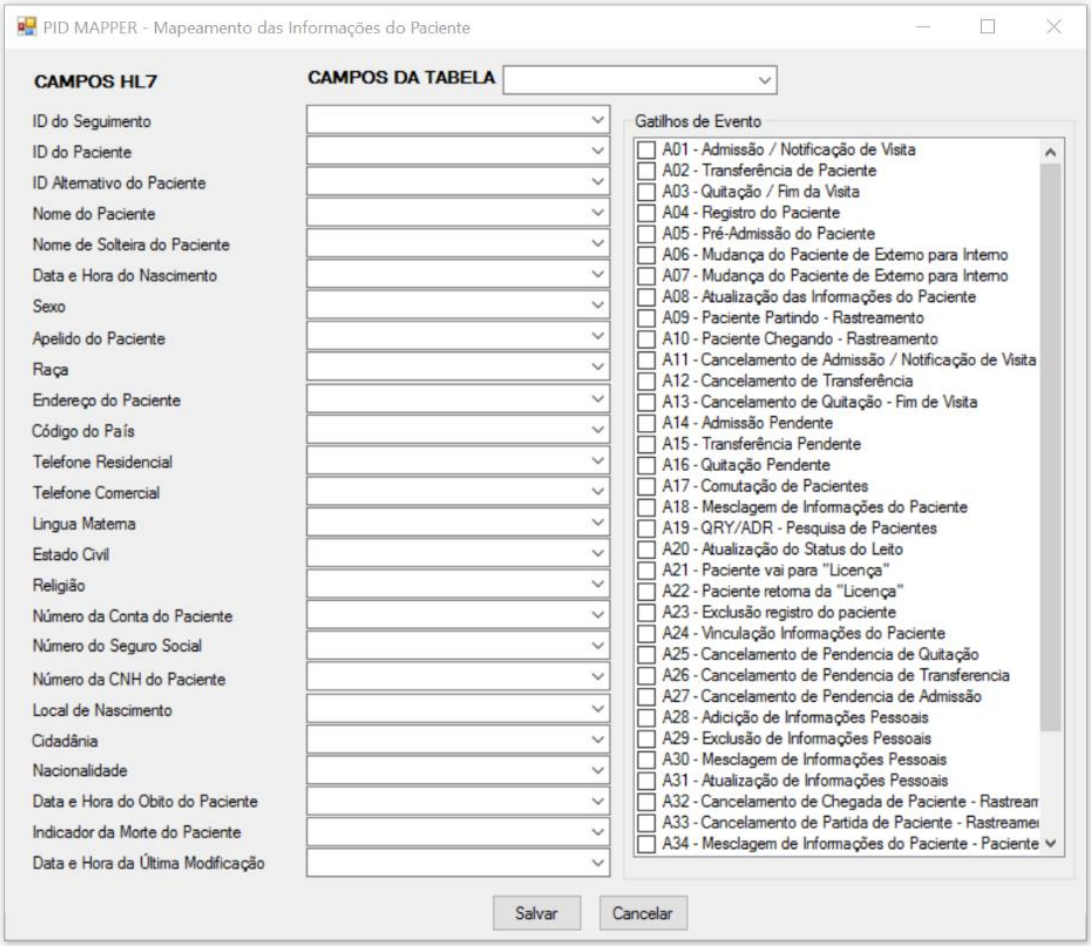


Figura 4. Interface gráfica de mapeamento das informações do paciente.

A sincronização dos dados com as demais instituições é realizada da seguinte forma: as mensagens disponíveis no hCloud Middleware são exibidas na interface gráfica de sincronização do hCloud Client, e para que ocorra a interoperabilidade, o usuário deve acionar a funcionalidade que requisitará ao hCloud Middleware o início da transmissão das mensagens. As mensagens recebidas são armazenadas em uma base de dados local e ficarão disponíveis para consulta, edição e persistência no sistema de banco de dados da instituição de saúde. A Figura 5 mostra a funcionalidade de sincronização das mensagens disponíveis na nuvem.

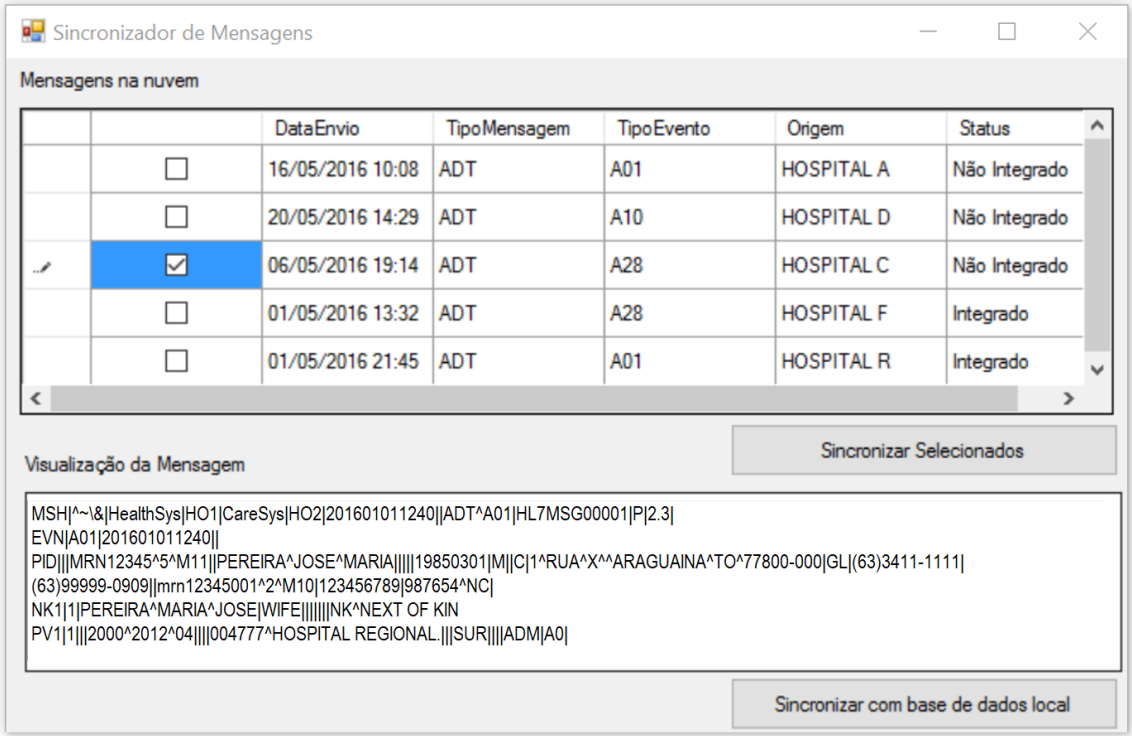


Figura 5. Funcionalidade de sincronização de mensagens.

Por meio da funcionalidade mostrada na Figura 5, é possível visualizar a mensagem HL7, bem como, verificar os detalhes a respeito da data do envio, tipo da mensagem (e.g., admissão), tipo do evento que gerou a mensagem (e.g., admissão/notificação de visita), instituição de origem e status da mensagem. Por fim, tem-se a opção ainda da sincronização individual ou grupos de mensagens.

5. Conclusão

Este artigo apresentou um middleware baseado no padrão HL7 capaz de interoperar os dados do Registro Eletrônico em Saúde (RES) por meio de um serviço em nuvem que codifica, persiste e sincroniza os dados do RES entre SIS. Como principais contribuições, destaca-se: i) a especificação de uma arquitetura que mostra os componentes de software e os seus relacionamentos; ii) a implementação de uma ferramenta e dos componentes que mapeiam e codificam as informações de um esquema de dados relacional em mensagens HL7; por fim, iii) exemplificou-se como é realizada a interoperabilidade dos dados por meio da ferramenta.

Existem duas vantagens principais da solução proposta. Primeiro, o middleware faz uso de uma arquitetura em nuvem o que diminui a necessidade de recursos computacionais para executar o serviço. Segundo, a partir de um esquema de dados relacional pode-se construir mensagens para troca de dados entre instituições de saúde utilizando o padrão HL7.

References

Beale, T. and Heard, S. (2007). openEHR - Architecture Overview. *The OpenEHR Foundation*, p. 1–79.

Bezerra, C., Araujo, A., Sacramento, B., Pereira, W. and Ferraz, F. (2015). Middleware For Heterogeneous Healthcare Data Exchange : A Survey. *ICSEA 2015 : The Tenth International Conference on Software Engineering Advances*, p. 409–414.

IBM (2016). IBM Message Broker 8. http://www-01.ibm.com/support/knowledgecenter/SSKM8N_8.0.0/com.ibm.healthcare.doc.

Ieee (2008). *Health informatics-Personal health device communication Part 10407: Device specialization - Blood pressure monitor*.

ISO (1996). Information technology -- Open Systems Interconnection (OSI) abstract data manipulation C language interfaces -- Binding for Application Program Interface (API).

ISO (2008). ISO 13606-2:2008 Health informatics Electronic healthcare record communication Part 2: Archetype interchange specification. International Organization for Standardization. http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=50119, [accessed on Jun 6].

Ko, L.-F. K. L.-F., Lin, J.-C. L. J.-C., Chen, C.-H. C. C.-H., et al. (2006). HL7 middleware framework for healthcare information system. *HEALTHCOM 2006 8th International Conference on e-Health Networking, Applications and Services*, p. 152–156.

Liu, X., Ma, L. and Liu, Y. (2012). A middleware-based implementation for data integration of remote devices. *Proceedings - 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD 2012*, p. 219–224.

Meta Healthcare (2015). Mirth Connect - HL7 Middleware. http://www.metahealthcare.com/solutions/mirth/, [accessed on Jun 21].

MITRE Corporation (2015). Project hData. http://www.projecthdata.org, [accessed on Jun 21].

Noumeir, R. and Pambrun, J.-F. (2010). Hands-on approach for teaching {HL7} version 3. *Information Technology and Applications in Biomedicine ({ITAB)}, 2010 10th {IEEE} International Conference on*, p. 1–4.

Petry, K., Marien, P. and Lopes, A. (2005). Modelos Para Interoperabilidade De Sistemas Hospitalares Utilizando Padrão Hl7.

Siemens (2016). Health Level Seven (HL7). http://www.healthcare.siemens.com/services/it-standards/hl7, [accessed on May 23].

Um Estudo Exploratório das Ferramentas de Código Aberto para a Replicação de Dados no PostgreSQL

Danilo S. de Carlo, Darlan F. S. Andrade, Rafael Liberato, André L. Schwerz

[1]Universidade Tecnológica Federal do Paraná (UTFPR)
Campus Campo Mourão - Departamento de Computação (DACOM)

{daniloc, darlanandrade}@alunos.utfpr.edu.br,

{liberato, andreluis}@utfpr.edu.br

Abstract. *Data distribution is indispensable to keep applications available and accessible when dealing with a high volume of requests. To meet this demand, PostgreSQL provides a wide variety of tools for data replication. However, understanding its characteristics and limitations is complex and time consuming. In this paper, we present an exploratory study about the features and limitations of the main open source tools for data replication in PostgreSQL, to reduce the complexity and time spent by database administrators in choosing the tool that meets their replication requirements.*

Resumo. *A distribuição de dados é indispensável para manter aplicações disponíveis e acessíveis ao lidar com um grade volume de requisições. Para atender essa demanda, o PostgreSQL fornece uma vasta variedade de ferramentas para replicação de dados. Entretanto, entender suas características e limitações é algo complexo e demanda tempo. Neste artigo, apresenta-se um estudo exploratório sobre os recursos e limitações das principais ferramentas de código aberto para a replicação de dados no PostgreSQL, buscando reduzir a complexidade e o tempo gasto por administradores de bases de dados na escolha da ferramenta que melhor preencha seus requisitos de replicação.*

1. Introdução

Um Banco de Dados Distribuídos (BDD) é uma coleção de vários bancos de dados interligados logicamente por meio de uma rede de computadores [Gupta et al. 2011]. Neste contexto, servidores de bancos de dados podem trabalhar juntos para que os dados de uma aplicação estejam sempre disponíveis e acessíveis mesmo em caso de falhas ou de um alto número de acessos simultâneos. Desta forma, em caso de falha em um dos servidores da rede, outros servidores podem suprir a demanda para que os dados mantenham-se disponíveis. Além disso, um número alto e distribuído de servidores e o balanceamento de cargas podem atender uma alta demanda de acessos simultâneos que a aplicação pode requerer. A distribuição dos dados entre os vários nós de um BDD é feita por meio da replicação ou fragmentação de dados [Heisler 2008].

Os Sistemas de Gerenciamento de Banco de Dados (SGBDs) relacionais mais conhecidos como o MariaDB [MariaDB 2019], SQL Server [Server 2019], PostgreSQL [Group 2019a], entre outros, possuem soluções para distribuir dados entre os servidores da rede. Bancos de dados NoSQL também possuem soluções para replicação

[Tauro et al. 2013], entretanto somente soluções relacionais são abordadas neste artigo. MariaDB provê alguns métodos de replicação como a replicação primário-réplica, replicação anel, replicação estrela e replicação de múltiplas fontes [MariaDB 2019]. SQL Server provê replicação transacional, replicação de mesclagem e replicação instantânea (*snapshot*) [Server 2019]. Em especial, foco deste estudo, o PostgreSQL possui diversas ferramentas e extensões para realizar o gerenciamento de BDD. Estas ferramentas combinam métodos de replicação e características de gerenciamento de dados distribuídos para atender demandas específicas.

Embora, o grande número de ferramentas do PostgreSQL disponíveis seja um fator positivo, compreender suas características e limitações não é uma tarefa trivial. A ampla variedade de alternativas para replicação, demanda conhecimento do projetista na escolha da solução mais satisfatória para o seu projeto. Por exemplo, em certos cenários pode ser mais importante manter os dados sempre consistentes no sistema do que prover uma alta disponibilidade ou vice-versa. Há meios de replicação que podem colocar a consistência dos dados em risco se não forem corretamente planejados.

Para auxiliar o projetista nesta escolha, apresentamos um estudo exploratório sobre os recursos, características e limitações das principais ferramentas de código aberto para a replicação de dados no PostgreSQL. O objetivo do estudo é levantar informações sobre as ferramentas para identificar quais são as mais eficazes em diferentes situações. Com isso, espera-se reduzir a complexidade e o tempo gasto por administradores de bases de dados na pesquisa e escolha da ferramenta que melhor preencha seus requisitos de replicação.

Este artigo está dividido como se segue. As principais características sobre replicação e bancos de dados distribuídos são descritas na Seção 2. A lista de ferramentas que apoiam a replicação no PostgreSQL é apresentada na Seção 3. Os principais resultados desta pesquisa estão na Seção 4 e os trabalhos relacionados estão na Seção 5. Por fim, as considerações finais são descritas na Seção 6.

2. Conceitos de replicação de banco de dados

A dificuldade fundamental em BDD é a sincronização dos dados entre os diferentes servidores. Qualquer escrita em um servidor precisa ser propagada para todos os outros servidores, para que futuras requisições de leituras nestes servidores retornem resultados consistentes. Normalmente, os servidores que recebem as alterações e replicam os dados são chamados de primários e os servidores que recebem esses dados são chamados de réplicas.

Uma replicação assíncrona é utilizada quando a propagação de operações de escrita ocorre em momentos oportunos (em um certo intervalo de tempo) para os outros servidores na rede [Heisler 2008]. Suas principais desvantagens estão na detecção tardia de possíveis erros e conflitos, e no tempo em que os servidores permanecem desatualizados até que as alterações sejam transmitidas. Por outro lado, na replicação síncrona, a propagação das operações de escrita para todos os servidores ocorre assim que elas chegam, e nenhuma transação é considerada finalizada até que todos os servidores tenham realizado aquela transação [Heisler 2008]. Suas principais desvantagens são uma queda significativa no desempenho e, na indisposição de algum servidor, podem ocorrer possíveis travamentos ou cancelamentos de operações de escrita. Nenhuma das soluções

elimina o impacto do problema de sincronização para todos os cenários. Desta forma, há diversas características e formas de replicação que devem ser observadas pelas ferramentas que a apoiam. As principais são descritas a seguir.

Modelo de replicação: os modelos básicos de replicação são o **primário-réplica** e o **multi-primário** [Mazilu et al. 2010]. Ambos os modelos podem ser síncronos ou assíncronos. No modelo primário-réplica, todas as modificações de escrita são realizadas no servidor primário e distribuídas para as réplicas, utilizando algum dos métodos de replicação. Tanto na solução assíncrona quanto na solução síncrona, os conflitos de sincronização não são comuns. No modelo multi-primário, cada servidor trabalha de forma independente, recebendo instruções de leitura e de escrita. Na solução assíncrona, o envio das modificações é realizado para todos os servidores periodicamente [Group 2019a]. Geralmente, o método assíncrono resulta em bastantes conflitos, que podem ser resolvidos manualmente ou por meio de regras de resolução. Por outro lado, no método síncrono, podem haver bloqueios prolongados e queda de desempenho. Há, ainda, outros modelos de replicação, como a replicação em anel ou a replicação em estrela, porém são variações derivadas da combinação dos modelos primário-réplica e multi-primário.

Balanceamento de carga: o balanceamento de carga pode ser divido em **balanceamento de leitura** e **balanceamento de escrita**. O balanceamento de leitura é o mais simples e pode ser executado desde que haja algum tipo de replicação. As requisições de leitura enviadas ao BDD são divididas entre os vários servidores da rede de forma que nenhum fique sobrecarregado. O balanceamento de escrita é mais complexo e dependente da forma que a replicação foi aplicada. BDD que possuem apenas um servidor primário são inaptos a executar o balanceamento de escrita, visto que é restrito ao servidor primário a execução das instruções de alteração. Sistemas que permitem a escrita em mais de um servidor podem executar o balanceamento de escrita, porém deve-se ficar atento aos possíveis conflitos e ao desempenho das sincronização dessas modificações.

Tolerância a falhas (*failover*): é inevitável que em um BDD servidores venham eventualmente a ser desligados da rede, seja devido a falhas ou a manutenções preventivas. Quando isso ocorrer, o sistema dever ser capaz de automaticamente redirecionar o fluxo dos dados ou promover servidores para manter o BDD ativo e disponível. Ferramentas de replicação devem automatizar o tratamento de falhas reduzindo intervenções humanas.

Replicação Parcial: replicar todos os dados de um BD nem sempre é algo desejável. A possibilidade de replicar tabelas específicas ou, até mesmo, partes de tabelas é um recurso útil em cenários de distribuição de dados.

Envio do *Write-Ahead Log (WAL)*: esse método de replicação, também conhecido por replicação física, consiste na replicação dos registros do WAL (*write-ahead log*), encaminhado do servidor primário para as diferentes réplicas [Group 2019a]. Pode ser realizado de forma síncrona ou assíncrona, não permitindo replicações parciais.

Replicação lógica: este método de replicação envia as alterações em um alto nível (*database/object level*) sendo mais seletivo que o WAL. Ele replica uma base de dados por vez, enviando apenas alterações confirmadas dos registros. Este método permite a replicação parcial da base de dados, e os dados podem fluir em múltiplas direções [Severalnines 2018].

Replicação baseada em *triggers*: neste método de replicação as modificações disparam gatilhos (*triggers*) que as transmitem assincronamente para os servidores réplicas [Group 2019a]. Além disso, como a atualização das réplicas é feita de forma assíncrona, um tratamento de falhas é necessário para não ocorrer perda de dados.

Fragmentação de dados: este método de replicação particiona a tabela em vários fragmentos (vertical ou horizontal), chamados *sets*, e os distribui entre os servidores. Cada servidor pode modificar apenas o seu *set* da tabela [Group 2019a]. É um método que divide os dados em quantidades menores, requer menor processamento das consultas e armazena tamanhos menores de índices.

Execução de instruções paralelas em múltiplos servidores (MPP): muitas das soluções existentes permitem a vários servidores tratar várias instruções SQL, mas esta permite vários servidores tratarem uma única instrução SQL simultaneamente, para a completar de forma rápida [Group 2019a]. É comumente utilizado com a fragmentação de dados, em que cada servidor executa a instrução para sua porção de dados, e retorna os resultados para um servidor central que combina os resultados e, então, retorna ao usuário [Group 2019a]. Este método diminui consideravelmente o tempo de resposta para quantidades massivas de dados.

Dado uma visão geral sobre os principais conceitos sobre replicação de dados, as diversas ferramentas de código aberto que apoiam a replicação no PostgreSQL serão abordadas a seguir.

3. Ferramentas de replicação do PostgreSQL

As diversas ferramentas de código aberto para replicação de dados no PostgreSQL foram selecionadas por meio da documentação do [Group 2019a] e de artigos científicos que utilizam dessas ferramentas para experimentos entre outras finalidades. Suas principais características estão na Tabela 1.

Ferramenta	Maturidade	Licença	Versões suportadas do PostgreSQL	Versão atual
PgCluster	Descontinuado	BSD	8.0	1.3
Mammoth	Descontinuado	BSD	8.3	1.8
Bi-Directional Replication (BDR)	Estável	BSD	9.4	1.0.65
pglogical	Estável	PostgreSQL	9.4 a 11	2.2.1
rubyrep	Inativo	MIT	Não encontrado	2.0.1
Pgpool-II	Estável	BSD	7.3 a 11	4.0.1
Slony-I	Estável	BSD	8.3 a 10	2.2.7
Bucardo	Estável	BSD	8.1 a 11	5.5.0
Postgres-XL	Estável	PostgreSQL	9.5 a 10	10R1
Citus	Estável	AGPL	9.5 a 10	7.5

Tabela 1. Ferramentas de replicação do PostgreSQL com suas características

No desenvolvimento deste trabalho, consideramos ferramentas que apoiam as versões ativas do PostgreSQL, 9.3 a 11, estão estáveis e não estão descontinuadas ou

inativas (sem atualização no último ano). As ferramentas PgCluster [PgFoundry 2009] e Mammoth [Bishop 2010] estão descontinuadas, portanto não serão abordadas neste trabalho. A ferramenta Bi-Directional Replication [2ndQuadrant 2019b] possui uma versão 3.0 proprietária que tem como dependência a extensão pglogical [2ndQuadrant 2019a], adicionando recursos a essa ferramenta para poder realizar replicações multi-primário e algumas outras operações. Embora sua versão 1.0.65 seja de código aberto, ela funciona apenas em uma versão modificada do PostgreSQL 9.4, e por isso não será avaliada neste artigo. A ferramenta pglogical [2ndQuadrant 2019a] possui diversas restrições em sua replicação, principalmente em redes de com muitas máquinas onde o fato da configuração das relações de *provider-subscriber* só pode ser realizada uma máquina de cada vez. Nesse cenário, também só pode haver um índice, ou uma restrição (*constraint*), ou uma chave primária. Dado estas restrições, o pglogical não será avaliado neste artigo. A ferramenta rubyrep [Lehmann 2017] está inativa, com atualizações esporádicas, a última datando da metade de 2017, e também não será considerada para as comparações. Por fim, a ferramenta Pgpool-II também não será avaliada, pois sua documentação recomenda outros meios para realizar a replicação, como a replicação nativa do PostgreSQL ou a ferramenta Slony-I, sendo a replicação nativa do próprio Pgpool-II a menos recomendada [PgFoundry 2019].

A replicação nativa do PostgreSQL e as ferramentas: Slony-I, Bucardo, Postgres-XL e Citus serão abordadas neste artigo e estão explanadas a seguir.

3.1. Replicação nativa do PostgreSQL

A replicação nativa do PostgreSQL [Group 2019a] utiliza o método de replicação WAL (Write-Ahead-Logging), em que é realizado a transmissão e cópia do log do banco de dados para os outros servidores, de modo que os dados entre eles fiquem consistentes. Inicialmente assíncrona em sua primeira implementação na versão 9.0 do PostgreSQL, ela pode ser configurada para funcionar de maneira síncrona a partir da versão 9.1. A partir da versão 10 do PostgreSQL replicação lógica passou a ser apoiada nativamente. A realização de cascateamento e de promoção de nós são possíveis, assim permitindo tolerância a falhas com uma configuração manual, envolvendo reiniciamento do nó primário problemático para resolução de problemas e reinserção.

3.2. Slony-I

O Slony-I [Group 2019b] é uma ferramenta de replicação assíncrona que utiliza a replicação primário-réplicas baseada em *triggers*. A ferramenta suporta replicação parcial sendo possível replicar apenas algumas tabelas do BD. Também é possível que diferentes nós repliquem diferentes tabelas. Cascateamento, promoção de nós e tolerância a falhas de forma automatizada também são suportados pela ferramenta. A ferramenta funciona com réplicas em diferentes versões do PostgreSQL e em diferentes sistemas operacionais. O Slony possui configuração complexa, sendo necessário adicionar, classificar e gerenciar todos os nós por meio de *scripts bash*.

3.3. Bucardo

Uma ferramenta de replicação assíncrona, multi-primário e primário-réplicas baseada em *triggers*. Aceita qualquer número de fontes (*primary*) e alvos (*replica*), replicação parcial e por demanda [Bucardo 2019]. Essa ferramenta permite lidar com a falha de nós da rede

de duas maneiras. No caso de falha em uma replica, realiza-se o redirecionamento do fluxo para outro nó. No caso de falhas de um servidor primário, há uma pequena sequência de passos a serem seguidos, sendo necessário uma resolução de conflito personalizada para casos de replicação multi-primário. Tolerância a falhas não é um objetivo primário do Bucardo [Bucardo 2019], de forma que nenhum dos tratamentos citados acima funcionam nativamente nem são configurados de forma automática.

3.4. Postgres-XL

A ferramenta Postgres-XL [Postgres-XL 2019] realiza a replicação de forma síncrona e balanceamento de leitura e escrita, utilizando MPP de forma transparente. O Postgres-XL possui dois componentes que lidam com os dados: o *Coordinator* e o *Datanode*. O *Coordinator* é uma interface que recebe as declarações em SQL, as analisa e planeja, para então determinar quais *Datanodes* serão envolvidos na transação, enviando uma plano serializado para cada componente envolvido.

3.5. Citus Community

A ferramenta Citus [Citus Data 2019] realiza replicação síncrona e assíncrona, apresentando também um recurso de tabelas distribuídas e MPP. Ela possui um nó coordenador que serve os nós *workers*. O coordenador possui apenas os metadados dos *workers*, que por sua vez armazenam as tabelas de dados.

A forma como a ferramenta Citus lida com o balanceamento de carga e com a fragmentação de dados facilita a inclusão de novos nós na rede. Quanto ao tratamento de falhas, existem algumas opções. Para tratar falhas em *workers*, Citus recomenda: (i) para sistemas OLTP, com grandes cargas de trabalho, baseia-se em habilitar a replicação nativa do PostgreSQL para substituir temporariamente o nó; e (ii) para sistemas com cargas de trabalho de anexação (leituras), baseia-se em habilitar a replicação de fragmentos do Citus no nó, garantindo os seus dados sejam replicados para outros nós. No caso do nó coordenador, que são comparativamente menores e menos manipuladas, a replicação nativa do PostgreSQL também pode ser utilizado, tornando simples e rápida a tarefa de substituição do nó em caso de falhas ou manutenções programadas. Servidores em *standby* são gerados utilizando WAL, de maneira que a alta disponibilidade é garantida.

4. Resultados

Existem diversos conceitos e características ligados a replicação de banco de dados, resultando em vários métodos de replicação. As ferramentas de replicação de banco de dados analisadas implementam um ou mais métodos de replicação aliados a características e/ou facilidades provenientes da própria ferramenta. Um resumo da comparação entre as características e métodos de replicação está exposto na Tabela 2.

Modelo de replicação: a ferramenta Bucardo apoia primário-réplica e multi-primário. A replicação nativa do PostgreSQL e a ferramenta Slony-I apoiam apenas replicação primário-réplica e as ferramentas Postgres-XL e Citus apoiam apenas multi-primário.

Balanceamento de Carga: uma vez que para realizar o balanceamento de leitura basta haver dados replicados, todas as ferramentas analisadas permitem esse tipo de balanceamento. A ferramenta Citus, em especial, pode apresentar melhores resultados nesta tarefa caso utilize a fragmentação de dados aliada ao processamento paralelo de instruções

Características	Ferramentas				
	Replicação nativa do PostgreSQL	Slony-I	Bucardo	Postgres-XL	Citus
Modelo de sincronismo	Síncrono e Assíncrono	Assíncrono	Assíncrono	Síncrono e Assíncrono	Síncrono
Modelo de replicação	Primário-Réplica	Primário-Réplica	Primário-Réplica e Multi-Primário	Multi-Primário	Multi-Primário
Balanceamento de carga	Apenas Leitura	Leitura e Escrita*	Leitura e Escrita	Leitura e Escrita	Leitura e Escrita
Tolerância a falhas automatizada		✓			✓
Replicação Parcial		✓	✓		✓
WAL	✓				
Replicação Lógica	✓				
Replicação por meio de *triggers*		✓	✓		
Fragmentação de dados					✓
MPP				✓	✓

Tabela 2. Características e métodos de replicação ofertadas pelas ferramentas analisadas.

SQL. O balanceamento de escrita, no entanto, é mais complexo e pode facilmente gerar conflitos e bloqueios em excesso. A única ferramenta analisada que não permite balanceamento de escrita é a replicação nativa do PostgreSQL. A ferramenta Slony-I permite que haja mais de um servidor primário na rede, sendo que cada um deles deve ser responsável por tabelas diferentes. Este recurso permite realizar balanceamento de escrita até certo nível, ao deixar servidores diferentes responsáveis pela escrita de tabelas diferentes. Não é possível, no entanto, realizar o balanceamento entre instruções de escrita na mesma tabela. As demais ferramentas viabilizam o balanceamento de escrita na mesma tabela, porém pode haver vários travamentos que podem causar lentidão. As ferramentas Postgres-XL e Citus, por utilizarem o processamento paralelo de instruções SQL, podem ter o melhor desempenho neste balanceamento.

Tolerância a falhas automatizada: a rápida recuperação de uma falha é uma característica importante para manter a disponibilidade da aplicação. As ferramentas Slony-I e Citus permitem que tais recuperações sejam feitas de forma automatizada, desde que configurado anteriormente. No Slony-I deve-se informar qual nó será promovido para primário em caso de falhas. A Citus replica os fragmentos de um nó para uma quantidade previamente estipulada de outros nós na mesma rede. Quando um nó falha, a ferramenta automaticamente redireciona o fluxo daquele nó para outro nó que possua os dados solicitados. As demais ferramentas apresentam recursos primitivos para tratamento de falhas, que envolvem intervenção manual e retrabalho.

Replicação parcial: dentre as ferramentas analisadas, Slony-I, Bucardo e Citus permitem a replicação parcial de dados. As ferramentas Slony-I e Bucardo permitem que apenas tabelas específicas sejam replicadas para a rede de banco de dados distribuídos. A ferramenta Citus permite fragmentar os dados de forma que é possível replicar até mesmo apenas determinadas linhas ou colunas de uma tabela.

WAL: este método de replicação implementado pela ferramenta nativa do PostgreSQL

executa a replicação de forma assíncrona ou síncrona. As demais ferramentas analisadas não apresentam este tipo de replicação implementado.

Replicação lógica: a partir de sua versão 10, o PostgreSQL disponibiliza nativamente a replicação lógica, porém este recurso ainda não está apto a lidar com a replicação de grandes quantidades de dados [Severalnines 2018]. A ferramenta pglogical implementa este método; entretanto, não foi considerada para esta avaliação devido aos motivos citados anteriormente.

Replicação baseado em *triggers*: as ferramentas Slony-I e Bucardo são as únicas a implementar este método de replicação. É um método assíncrono e pode mostrar dados infiéis ao acessar os nós réplicas caso haja alterações que ainda não foram aplicadas a eles. Ambas as ferramentas utilizam tabelas auxiliares para manter o controle das replicações.

Fragmentação de dados: a ferramenta Citus implementa este modelo de replicação de fragmentação de dados, em que divide-se linhas e/ou tabelas entre os nós da malha de replicação. Por ser uma replicação focada em desempenho, recomenda-se um servidor com uma cópia de todos os dados, sem fragmentação, seja mantido para casos de falha. Manter cópias dos fragmentos de um nó da rede em outros nós da rede também é uma opção. A ferramentas Citus é a única a implementar este método de replicação.

MPP: este modelo de replicação faz com que a malha de replicação se torne um *cluster*, paralelizando a execução e concentrando os resultados em um servidor central. Isso resulta em um aumento de desempenho, principalmente para dados massivos. Seu desempenho é maximizado quando combinado com a fragmentação de dados. A ferramenta Postgres-XL implementa este modelo de replicação, tal como a ferramenta Citus.

5. Trabalhos Relacionados

A replicação de dados é um tema abrangente e amplamente discutido na área de Banco de Dados. Um levantamento de ferramentas comerciais e replicação é apresentado em [Moiz et al. 2011], porém o PostgreSQL não foi o foco principal. Detalhes sobre o processo da replicação seus conceitos e contextos são apresentados em [Wiesmann et al. 2000]. Entretanto, não faz comparativo entre os métodos de replicação existentes. Há um interessante estudo comparativo das ferramentas Slony-I e pgpool-II apresentado em [Partio 2007]. Neste trabalho, realiza-se uma breve descrição uma avaliação comparativa de desempenho, com ênfase no balanceamento de leitura. Em [Mauchle 2008], realiza-se um breve comparativo entre a replicação de dados no MySQL e no PostgreSQL, citando algumas ferramentas de replicação, como Slony-I e Bucardo.

6. Considerações Finais

O PostgreSQL possui diversas ferramentas que possibilitam replicação dos dados; entretanto, determinar quais delas é mais adequada para cada situação pode ser algo complexo. Neste trabalho, realizou-se uma estudo exploratório das características relevantes das principais de ferramentas de replicação do PostgreSQL.

Dentre todas soluções analisadas, a ferramenta nativa do PostgreSQL é a que tem menos recursos. Embora possibilite tanto a replicação síncrona quanto a assíncrona, ela não permite balanceamento de escrita nem replicação parcial. É uma solução de replicação para casos simples que não lidem com grandes volumes de dados, nos quais a

tolerância a falhas de forma automática não se faz necessária. O Slony-I é uma solução que abrange dos casos mais simples (backup), aos mais complexos (balanceamento de carga de uma grande aplicação). Trabalha apenas de forma assíncrona e possui alguns recursos como a tolerância a falhas automatizada e a replicação parcial das bases de dados. Um diferencial interessante do Slony-I é que ele permite que haja mais de um nó primário no BDD, desde que cada nó primário seja responsável pela replicação de uma tabela diferente. Essa característica possibilita o balanceamento de escrita até certo nível, mesmo estando em um modelo primário-réplica.

Bucardo é uma ferramenta semelhante ao Slony-I, e suas diferenças moram em dois recursos: (i) o Bucardo realiza replicações multi-primário, o que não é possível no Slony-I; e (ii) a tolerância a falhas no Bucardo só existe de forma totalmente manual, diferentemente do Slony-I que possui uma forma automatizada. A ferramenta Bucardo permite ainda o balanceamento de escrita na mesma tabela, porém há falta de suporte a tratamento de falhas.

As duas ferramentas restantes, Postgres-XL e Citus, são recomendadas para bases com um grande volume de dados e vários servidores na rede de replicação. Ambas ferramentas transformam a rede de replicação em um *cluster*, aumentando o desempenho no processamento das consultas. A ferramenta Postgres-XL permite o balanceamento de carga e realiza sua replicação síncrona de forma paralela em todo o BDD. Ela não possui, entretanto, formas para replicação parcial ou tolerância a falhas automatizada. A ferramenta Citus também possibilita balanceamento de carga e realiza sua replicação de forma síncrona e paralela. A diferença entre estas ferramentas é que a Citus possui tolerância a falhas de forma automatizada e também possibilita a replicação parcial por meio da fragmentação da dados.

Como trabalho futuro, pretende-se realizar testes de desempenho com as ferramentas Slony-I e Citus por meio de *benchmarks* com o objetivo de averiguar as possibilidades de utilização dessas ferramentas em produção.

Referências

2ndQuadrant (2019a). pglogical — 2ndquadrant. Disponível em: https://www.2ndquadrant.com/en/resources/pglogical. Acesso em 17/01/2019.

2ndQuadrant (2019b). Postgres-bdr documentation. Disponível em: http://bdr-project.org/docs/stable/index.html. Acesso em 17/01/2019.

Bishop, S. (2010). Mammoth replicator. Disponível em: https://launchpad.net/mammoth-replicator. Acesso em 10/12/2018.

Bucardo (2019). Bucardo asynchronous postgresql replication system. Disponível em: https://bucardo.org/Bucardo. Acesso em 15/01/2019.

Citus Data, I. (2019). Citus documentation. Disponível em: https://docs.citusdata.com/en/v8.1/index.html. Acesso em 19/01/2019.

Group, P. G. D. (2019a). Postgresql. Disponível em: http://www.postgresql.org. Acesso em 16/03/2019.

Group, S. D. (2019b). Slony-i enterprise-level replication system. Disponível em: http://slony.info/. Acesso em 15/01/2019.

Gupta, S., Saroha, K., and Bhawna (2011). Fundamental research of distributed database. IJCSMS - International Journal of Computer Science and Management Studies, Vol. 11, Issue 02, Aug 2011 - Disponível em: https://pdfs.semanticscholar.org/f935/1f0cf3c4307dd76c85d6815e2a1b8095324b.pdf. Acesso em 16/03/2019.

Heisler, D. A. (2008). Estudo de algoritmos e técnicas de teplicação de banco de dados em software livre. Disponível em: https://www.univates.br/bdu/bitstream/10737/563/1/2008DanielAfonsoHeisler.pdf. Acesso em 06/02/2019.

Lehmann, A. (2017). rubyrep: Home. Disponível em: http://www.rubyrep.org. Acesso em 10/12/2018.

MariaDB (2019). Replication overview. Disponível em: https://mariadb.com/kb/en/library/replication-overview/. Acesso em 09/02/2019.

Mauchle, F. (2008). Database replication with mysql and postgresql. Disponível em: https://wiki.hsr.ch/Datenbanken/files/Mauchle_Replication_MySQL_Postgres_Paper.pdf. Acesso em 08/02/2019.

Mazilu, M. C. et al. (2010). Database replication. *Database Systems Journal*, 1(2):33–38.

Moiz, S. A., P., S., G., V., and Pal, S. N. (2011). Article: Database replication: A survey of open source and commercial tools. *International Journal of Computer Applications*, 13(6):1–8.

Partio, M. (2007). Evaluation of postgresql replication and load balancing implementations. Unpublished.

PgFoundry (2009). Pgfoundry: Pgcluster. Disponível em: http://pgfoundry.org/projects/pgcluster. Acesso em 10/12/2018.

PgFoundry (2019). Pgpool wiki. Disponível em: http://www.pgpool.net/mediawiki/index.php/Main_Page. Acesso em 11/01/2019.

Postgres-XL (2019). Open sourcescalable sql database cluster. Disponível em: https://www.postgres-xl.org. Acesso em 15/01/2019.

Server, S. (2019). Tipos de replicação. Disponível em: https://docs.microsoft.com/pt-br/sql/relational-databases/replication/types-of-replication. Acesso em 09/02/2019.

Severalnines (2018). An overview of logical replication in postgresql. Disponível em: https://severalnines.com/blog/overview-logical-replication-postgresql. Acesso em 08/02/2019.

Tauro, C. J., Patil, B. R., and Prashanth, K. (2013). A comparative analysis of different nosql databases on data model, query model and replication model. In *Proceedings of the International Conference on ERCICA*.

Wiesmann, M., Pedone, F., Schiper, A., Kemme, B., and Alonso, G. (2000). Understanding replication in databases and distributed systems. In *Proceedings 20th IEEE International Conference on Distributed Computing Systems*, pages 464–474.

Uma Análise de Soluções NewSQL

Ronan R. Knob[1], **Geomar A. Schreiner**[1],
Angelo A. Frozza[1,2] , **Ronaldo dos Santos Mello**[1]

[1]Departamento de Informática e Estatística (INE)
Universidade Federal de Santa Catarina (UFSC)
Florianópolis, SC – Brazil

[2]Instituto Federal Catarinense (IFC) - Campus Camboriu
Rua Joaquim Garcia, S/N – 88.340-055 – Camboriu (SC), Brasil

geomarschreiner@gmail.com, angelo.frozza@ifc.edu.br

ronanknob@grad.ufsc.br, r.mello@ufsc.br

Abstract. *Several applications have as requirements the need to handle large and heterogeneous data volumes as well as the support to handle thousands of OLTP transactions per second. Traditional relational databases (DBRs) are not suitable for these requirements. On the other hand, NoSQL DBs are able to deal with Big Data, but lacks the support to ACID properties. NewSQL is a new class of DBs that combines the support to OLTP transactions of BDRs with the high availability and scalability of NoSQL DBs. However, few works in the literature explore the differences among different NewSQL solutions. In this paper, we execute benchmark software to compare the most prominent NewSQL products analyzing the results. This analysis can be useful as a guide to future use of NewSQL technology.*

Resumo. *Diversas aplicações produzem e manipulam grandes volumes heterogêneos de dados, bem como necessitam lidar com um grande número de transações OLTP. Os tradicionais Bancos de Dados Relacionais (BDRs) não são adequados a este tipo de demanda. Já os BDs NoSQL, apesar do melhor gerenciamento de Big Data, não garantem as propriedades ACID. O movimento NewSQL visa suportar transações OLTP dos BDRs com uma arquitetura distribuída que oferece alta escalabilidade e disponibilidade, típica dos BDs NoSQL. Poucos trabalhos na literatura exploram as diferenças entre soluções NewSQL. Assim, este trabalho visa comparar alguns dos principais produtos NewSQL utilizando benchmarks de domínio. Esta análise contribui como um guia de referência para futuros usos da tecnologia NewSQL.*

1. Introdução

Os avanços em tecnologias Web e a proliferação de dispositivos móveis conectados à Internet gerou uma necessidade de tratamento de grandes quantidades de dados heterogêneos em curto espaço de tempo. Dados com esta natureza de gerenciamento são denominados *Big Data*. Um conjunto de novas aplicações, como sistemas financeiros e jogos *online*, lidam com um grande número de transações *OLTP* (*Online Transaction Processing*) executadas sobre Big Data [Stonebraker 2012]. Transações OLTP são, em

geral, transações de curta duração e que não processam grandes quantidades de dados. BDRs, apesar de empregados na manipulação eficiente de dados durante décadas, não são adequados ao tratamento de Big Data e transações OLTP com alta disponibilidade impostos por estas aplicações [Pavlo and Aslett 2016].

Motivado por esses desafios, surgiram novas soluções de BD denominadas BDs NoSQL (*Not Only SQL*). Estas soluções oferecem recursos como alta disponibilidade e escalabilidade, atrelados a uma arquitetura distribuída e com crescimento horizontal. Apesar de serem capazes de manipular grandes volumes de dados com alta disponibilidade, BDs NoSQL geralmente não suportam as tradicionais propriedades ACID que caracterizam transações OLTP.

Mais recentemente, o paradigma NewSQL surgiu com o propósito de combinar os benefícios do paradigma relacional com o tratamento de Big Data do paradigma NoSQL. Sistemas NewSQL são soluções modernas que buscam prover o mesmo desempenho escalável dos BDs NoSQL para cargas de trabalho OLTP com típico suporte completo a todas as propriedades ACID, como encontrado nos BDRs [Pavlo and Aslett 2016]. Um BD NewSQL deve considerar dois importantes fatores: *(i)* Um controle de concorrência de esquema *lock-free*; e, *(ii)* Uma arquitetura distribuída *shared-nothing* [Stonebraker 2012]. Apesar de compartilharem características gerais, como as citadas anteriormente, cada sistema NewSQL possui sua própria maneira de executar operações de manipulação de dados. Assim sendo, torna-se pertinente uma análise de soluções NewSQL através de *benchmarks* (protocolos de testes padronizados) capazes de mensurar desempenho a fim de verificar a eficácia das mesmas em situações reais.

Este trabalho tem como objetivo contribuir com a literatura acerca do paradigma NewSQL, comparando quatro soluções relacionadas: VoltDB, NuoDB, MemSQL e Cockroach. A comparação é realizada através da execução de dois *benchmarks* (YCSB e Votter), bem como uma análise dos resultados dos testes para estas soluções. A escolha das soluções considerou os seguintes critérios: o ranking das mesmas no site *DB-Engines*[1]; número de menções em Web sites e interesse nas buscas (via Google Trends[2]); a análise da pesquisa anual do Web site *Stack Overflow* no ano de 2017 [3]; e licença de uso gratuito que permitisse os testes. A seleção dos *benchmarks* levou em consideração aqueles que possuíam um foco maior em transações OLTP.

O restante deste artigo está organizado conforme segue. A Seção 2 apresenta os trabalhos relacionados, enquanto a Seção 3 descreve os BDs NewSQL selecionados. O ambiente experimental e os *benchmarks* escolhidos são descritos na Seção 4. Na Seção 5 são discutidos os resultados e, por fim, as considerações finais encontram-se na Seção 6.

2. Trabalhos Relacionados

Poucas iniciativas para comparação exclusiva entre BDs NewSQL são encontradas na literatura. Cabe ressaltar que existem alguns trabalhos que comparam BDs NewSQL e NoSQL [Grolinger et al. 2013, Hajoui et al. 2015, Gurevich 2015]. Porém, eles foram desconsiderados pois realizam uma análise de cunho teórico entre soluções de dois paradigmas diferentes, não comparando unicamente soluções NewSQL.

[1] `https://db-engines.com/en/ranking`
[2] `https://trends.google.com.br/trends/`
[3] `https://insights.stackoverflow.com/survey/2017`

O trabalho de [Pavlo and Aslett 2016] faz uma análise histórica de sistemas de gerência de banco de dados (SGBDs) NewSQL. Porém, o trabalho não possui experimentação. Já os trabalhos de Kaur [Kaur and Sachdeva 2017] e Oliveira [Oliveira and Bernardino 2017] relatam avaliações de desempenho entre produtos NewSQL. O trabalho de Kaur realiza uma comparação entre os SGBDs *NuoDB*, *Cockroach*, *VoltDB* e *MemSQL* utilizando métricas simples para avaliação que podem ajudar no processo de escolha de uma solução. Entretanto, elas não são métricas adequadas, pois não consideram um ambiente distribuído no qual soluções NewSQL são tipicamente utilizadas. Já o trabalho proposto por Oliveira avalia o desempenho dos SGBDs *VoltDB* e *MemSQL* utilizando o software de benchmark TPC-H. O TPC-H, porém, tem como foco transações OLAP, enquanto soluções NewSQL priorizam em seu desenvolvimento transações OLTP. Além disso, o trabalho de Oliveira também não leva em consideração um ambiente distribuído para realizar seus experimentos.

O trabalho proposto neste artigo se difere dos trabalhos de Kaur e Oliveira ao comparar soluções *NewSQL* utilizando dois softwares de *benchmark* focados em transações OLTP em um ambiente distribuído.

3. Bancos de Dados NewSQL

BDs NewSQL são uma nova classe de BD que oferecem o mesmo desempenho escalável dos BDs NoSQL enquanto garantem as tradicionais propriedades ACID dos BDRs [Pavlo and Aslett 2016]. Eles são geralmente BDs em memória principal que maximizam a vazão de dados, prevenindo custosos acessos em disco aos dados. Eles também possuem mecanismos de controle de concorrência que evitam o bloqueio de dados, possibilitando alta disponibilidade de dados. Além disso, são BDs nativamente distribuídos com arquiteturas e algoritmos otimizados para este ambiente. Apesar destas características em comum, cada solução NewSQL possui, evidentemente, uma implementação distinta. As subseções a seguir apresentam brevemente os quatro SGBDs NewSQL selecionados neste trabalho. Uma comparação mais detalhada entre as soluções pode ser encontrada em [Knob 2018].

3.1. VoltDB

O VoltDB[4] é um SGBD desenvolvido desde 2010 por uma empresa que carrega o mesmo nome. Ele é disponibilizado em versões *enterprise* e *community*, sendo esta última sob licença *GNU Affero General Public License* [VoltDB 2015].

O VoltDB possui grande ganho de desempenho por serializar o acesso a todos os dados, prevenindo o consumo de tempo de funções de *latching* e *logs* de transação, dentre outras. Ele possui uma arquitetura de *cluster* com replicação sob múltiplos servidores, que garantem escalabilidade, confiabilidade e alta disponibilidade dos dados. O VoltDB é compatível com a linguagem *SQL ANSI*, o que garante uma rápida curva de aprendizado dos usuários.

Na arquitetura do VoltDB, os *clusters* contêm uma fila de processamento, uma *engine* de execução e as tabelas com os dados indexados. A comunicação entre nodos é realizada quando há necessidade de processar uma consulta que necessita dados de

[4] `https://www.voltdb.com/`

múltiplas partições. Neste cenário, um dos nodos age como coordenador e distribui o trabalho necessário entre os demais nodos, coletando os resultados e completando a tarefa [VoltDB 2013].

3.2. NuoDB

O NuoDB[5] lançou sua primeira versão em 2014. O produto é disponibilizado sob licença proprietária e possui as versões *Community*, que é gratuita, *Professional* e *Enterprise*. A versão *Community*, utilizada no trabalho, inclui restrições de escalabilidade. A arquitetura do NuoDB na versão gratuita permite uma (1) instância de administração, um (1) *SM (Storage Management)* e três *TEs (Transaction Processsing)*. Já a versão paga garante ilimitados SMs e TEs. Os conceitos de SM e TE fazem parte da arquitetura do NuoDB e são explicados a seguir.

TE é a camada que recebe requisições SQL, sendo constituída de nodos em memória chamados *Transaction Engines (TENs)*. Quando uma aplicação faz requisições ao NuoDB, os TENs criam *caches* em memória para a carga de trabalho da aplicação. As requisições para arquivos que não estão em *cache* são alimentadas com *caches* de memória de outros TENs ou pela camada de gerenciamento de armazenamento. Já um SM é um nodo de processamento que possui componentes em memória e em disco rígido. Ele também oferece garantias de durabilidade dos dados. Múltiplos SMs podem ser usados para aumentar a redundância de dados.

O NuoDB torna-se escalável pela simplicidade de suas duas camadas. As capacidades do cluster são dimensionadas através dos TEs e SMs. Com essa modularidade, há também a opção de escolher um modo de replicação para cada novo BD criado.

3.3. CockroachDB

O CockroachDB[6] foi lançado em 2015. O projeto foi criado para ser um BD *open source* e distribuído, de forma que uma instância pode ser levantada em um computador pessoal comum e ajudar no processamento de requisições [Labs 2018]. O produto é distribuído nas versões *Core* e *Enterprise*, sendo a primeira gratuita. Diferente das demais soluções NewSQL, ele não utiliza armazenamento final em memória principal. Ao invés disso, é feito o aproveitamento de uma estrutura de *clocks* atômicos para escrita de blocos, que facilita o suporte às características ACID nas transações.

A arquitetura do CockroachDB converte comandos SQL em estruturas de dados *Key-Value (KV)*, que são extremamente rápidas de manipular. Essa arquitetura é composta por um *SQL Layer* que recebe as consultas via uma interface API. A camada SQL repassa a consulta para a camada *Transaction Layer*. Esta camada gera um plano de execução para as requisições SQL. O plano é passado para a camada *Distribution Layer*, responsável por manter um mapa com os pares de chaves *KV*, um repositório que descreve todos os dados do *cluster* e sua localização. O mapa é dividido e distribuído em intervalos, de modo que a consulta às chaves não fique centralizada em um nodo.

O armazenamento efetivo dos dados é realizado pelo *Storage Layer*. Cada instância do BD deve possuir ao menos um *store*, que é o espaço no qual o processo

[5] `https://www.nuodb.com/`
[6] `https://www.cockroachlabs.com/`

do BD lê e escreve dados no disco. Os dados são guardados e manipulados através da API *RocksDB*[7], que mantém os pares KV no disco.

3.4. MemSQL

O MemSQL[8] teve sua primeira versão lançada em 2013. Ele é distribuído em duas versões. A *Developer*, gratuita, não é recomendada para uso em produção e tem recursos limitados. Já a versão *Enterprise* (paga) possui todas as funcionalidades. O MemSQL se categoriza como um BDR distribuído que suporta transações e análises em tempo real.

A estrutura do MemSQL é composta de duas camadas: os *nodos agregadores* e os *nodos folha* [MemSQL 2018]. Os nodos agregadores funcionam como roteadores de consulta e atuam como um *gateway* no sistema distribuído. Eles armazenam apenas metadados e dados de referência, distribuindo as consultas nos nós folha. Estes nodos também são responsáveis por agregar os resultados a serem enviados de volta ao cliente. Já os nodos folha armazenam e computam as tarefas. Os dados são automaticamente distribuídos através dos nodos folha, em partições sobre as quais as consultas são executadas de forma paralela.

A comunicação entre os nodos é realizada via comandos SQL sob um protocolo MySQL. A proporção de nodos agregadores e nodos folha determina a capacidade e o desempenho do cluster, que pode variar conforme a aplicação.

4. Ambiente Experimental

Os experimentos propostos para as soluções NewSQL escolhidas necessitaram de uma infraestrutura padronizada, bem como uma configuração de *cluster* específica para cada uma delas. Todas elas foram instaladas e configuradas de maneira padrão (sem nenhuma otimização) em um *cluster* com três nodos físicos. Cada um dos nodos possui um processador Intel® Core™ i5-7200 (4 núcleos físicos de $2,50$ GHz), com Memória RAM 8GB (DDR3 1333Mhz), disco rígido de 320GB (5400 RPM) e sistema operacional Xubuntu 16.04 Server LTS 64 bits. Devido a problemas de configuração, o *VoltDB* foi instalado no *cluster* utilizando *Docker*. Já os demais produtos foram instalados diretamente na máquina. Vale ressaltar que, apesar de todos os produtos terem sido instalados nas mesmas três máquinas, apenas um deles estava ativo durante os testes. Os nodos do *cluster* foram conectados via *Ethernet* (100 Mbps) sem acesso à rede externa.

Os experimentos foram realizados com o uso da ferramenta *OLTP-Bench* [Difallah et al. 2013], que foi executada em um nodo externo ao *cluster*. A *OLTP-Bench* é uma suíte de *benchmarking* que traz suporte a vários SGBDs comerciais, assim como uma boa carga de *benchmarks* distintos. Basicamente, a ferramenta gera uma fila de transações para execução de acordo com a especificação do *benchmark* escolhido e um arquivo de configuração fornecido pelo usuário. Esta fila é executada paralelamente por um número de *workers* (que emulam usuários) configurado no arquivo de entrada. Durante a execução dos testes, o *OLTP-Bench* coleta as estatísticas de execução retornando um arquivo com esses valores.

Considerando as características dos BDs NewSQL, foram selecionados *benchmarks* para BDRs com foco em OLTP, ou seja, capazes de simular um cenário com

[7] `https://rocksdb.org/`
[8] `https://www.memsql.com/`

transações simples, porém, em grande quantidade. Os *benchmarks Yahoo! Cloud Serving Benchmark (YCSB)* e *Voter* foram escolhidos com base nestes critérios, sendo apresentados nas próximas subseções.

Dentre as métricas disponibilizadas no arquivo de saída da *OLTP-Bench*, foram selecionadas duas para análise neste trabalho: *(i)* a taxa de transações executadas no tempo (*Throughput*); e, *(ii)* a latência das transações, através da análise da média geral das latências, bem como a análise dos percentis 90 e 99. Para o *benchmark* YCSB também foi realizada a análise de média das latências por tipo de transação. O desvio padrão das amostras observadas também foi calculado para ajudar a explicar as observações.

4.1. YCSB

O YCSB é uma aplicação geradora de carga de trabalho e um pacote com cargas padrão que cobrem interessantes partes da avaliação de desempenho, como cargas de leitura e escrita intensa, varredura de tabelas, dentre outras. Cada carga representa um misto de operações de leitura e escrita com diferentes volumes de dados e número de requisições.

A estrutura de teste consiste em uma única tabela, denominada *usertable*, com N campos. Cada registro é identificado por uma chave primária (algo como "user234123") e cada campo é nomeado como *field0*, *field1*, ..., *fieldN*. Os valores dos campos são strings randômicas de caracteres ASCII de tamanho aleatório. Os parâmetros *número de campos* (N) e *fator de escala* (F) são informados *a priori*.

A execução de um teste realiza muitas escolhas aleatórias, como as operações que serão feitas (*Insert, Update, Read* ou *Scan*), qual registro ler ou escrever, e quantos registros examinar. Essas decisões são governadas por distribuições randômicas. Nos parâmetros definidos para a execução são configurados fatores relacionados à escala de volume da base de teste e os dados da carga de trabalho. Neste trabalho foi utilizado o fator de escala 1000, 64 usuários virtuais emulados para manipulação (64 conexões simultâneas) e um limite do teste de 300 segundos. O volume total de dados gerado foi de $18,2$ GB.

4.2. Voter

O *Voter* é um *benchmark* baseado em um software utilizado em um programa de talentos exibido em televisão no Japão e no Canadá. Os usuários ligam para votar no seu candidato favorito. Ao receber uma ligação, a aplicação invoca a transação que atualiza o número total de votos de cada participante. Os votos feitos por cada usuário são armazenados em um BD e possuem um limite máximo configurável. Uma transação em separado é periodicamente invocada para computar os votos totais durante o programa.

Este *benchmark* foi desenvolvido com o intuito de saturar o BD com pequenas transações, todas atualizando um pequeno número de registros. A arquitetura do *Voter* possui três tabelas que guardam dados sobre os candidatos e o usuário que está ligando. Além disso, existem duas *views* que são consultadas para atualizar o status no programa de televisão. Neste trabalho foram utilizados como parâmetros o fator de escala (1000) e o número de usuários virtuais emulados (64). O volume de dados gerado foi de $2,6$ GB.

5. Análise dos Resultados

Esta seção apresenta os resultados obtidos com os *benchmarks* definidos na seção anterior.

5.1. *Benchmark* YCSB

Os resultados obtidos com o *benchmark YCSB* foram coletados a partir de 5 execuções de teste para cada solução NewSQL. O gráfico da Figura 1 mostra os resultados para a primeira métrica: número de transações executadas por segundo.

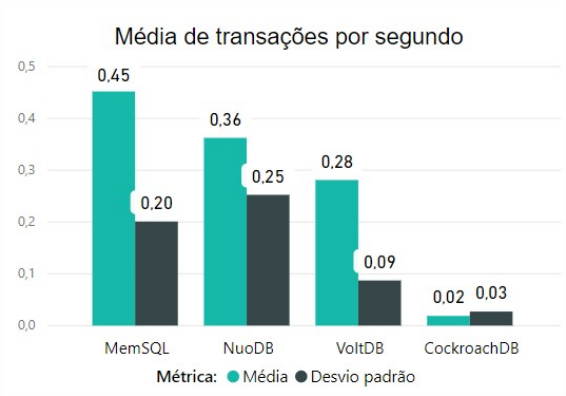


Figura 1. Médias de transações por segundo obtidas no *benchmark YCSB*.

Uma análise do gráfico demonstra que o *MemSQL* apresentou melhores resultados, executando em média $0,45$ transações a cada segundo do teste, à frente do NuoDB, que executou $0,36$. O *VoltDB*, por sua vez, executou $0,28$ transações por segundo em média e, por último, tem-se o *Cockroach*, executando $0,02$ transações por segundo em média. O desvio padrão mostra que o grau de dispersão entre os resultados foi baixo, ou seja, houve uma boa uniformidade na execução. As maiores discrepâncias de execução ficaram por conta do *MemSQL* e do *NuoDB*. A diferença considerável visível no Figura 1 do produto *CockroachDB* para os concorrentes pode ser explicada comparando as demais métricas, apresentadas a seguir.

Conforme descrito anteriormente, outra métrica considerada é a média de latência das transações. Os resultados para esta métrica são mostrados no gráfico da Figura 2.

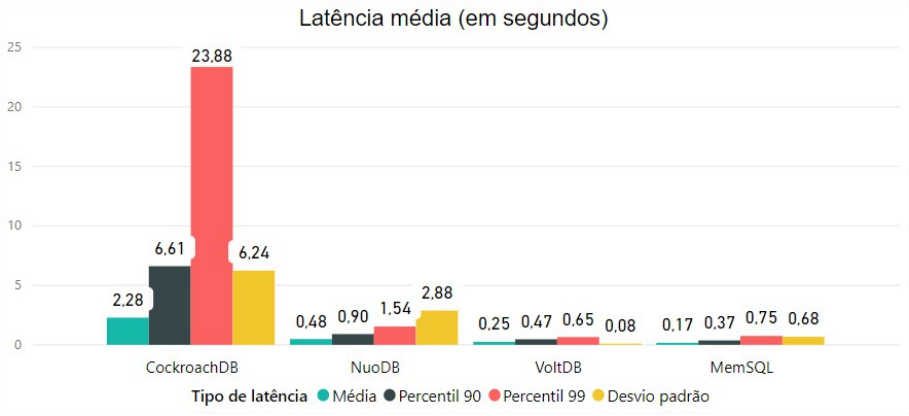


Figura 2. Médias de latência por segundo obtidas no *benchmark YCSB*.

Como se pode observar, o *CockroachDB* também apresentou maior latência média nas transações, quando não se considera o tipo de transação. O desvio padrão mostra que o grau de dispersão entre os resultados foi alto para *CockroachDB* e *NuoDB*, indicando que

estes produtos tiveram uma grande discrepância de valores, diferente dos outros produtos que mostraram uma execução mais uniforme. Em relação à média, *NuoDB* e *VoltDB* sucederam o *CockroachDB* no *ranking*, com resultados parecidos entre si. A melhor latência média ficou por conta do *MemSQL* ($0,17$ por segundo).

Uma análise dos percentis 90 e 99 mostra que uma quantidade expressiva das transações com maior latência se encontra em um pequeno número de transações situadas nesses pontos. O caso mais evidente é o *CockroachDB* (Figura 2), para o qual há uma grande diferença entre a média de $2,28$ segundos na média de latências e $23,88$ segundos na média de latências no percentil 99.

Os resultados apresentados pelo *CockroachDB* para latência média e média de transações por segundo são discrepantes com relação aos demais produtos. Isto se deve ao fato que o armazenamento principal do CockroachDB não é em memória principal, como os demais produtos, e sim em estruturas *KV* em disco. Este BD utiliza um percentual da memória principal para *cache*, mas utiliza grande parte de suas operações baseadas em disco, o que gerou as latências e a taxa de vazão observadas.

Discrepâncias também são notadas para o *NuoDB*. Tanto na média de transações por segundo, quanto na análise de latência média, o produto mostra um grau de dispersão um pouco elevado. Tais resultados tendem a ocorrer tendo em vista as limitações da versão gratuita, na qual é possível utilizar o armazenamento em apenas um dos nodos do *cluster*, mesmo que seja possível utilizar o componente de execução de transações em três nodos. O armazenamento em um *storage* único pode aumentar a latência da execução, visto que os nodos executam parte do teste em um nodo, transferindo todo o resultado para o nodo que mantém os dados.

5.2. *Benchmark* Voter

Da mesma forma que o *YCSB*, os resultados no *benchmark Voter* foram coletados com dados de 5 execuções do teste para cada solução. O gráfico apresentado na Figura 3 mostra o resultado da primeira métrica: número de transações executadas por segundo.

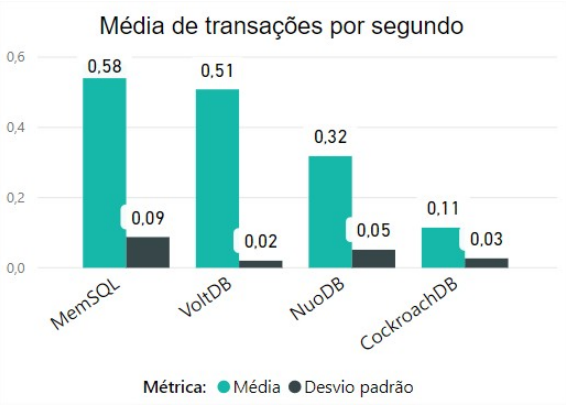


Figura 3. Médias de transações por segundo obtidas no *benchmark Voter*.

Uma análise do gráfico mostra novamente que o *MemSQL* apresenta os melhores resultados, executando em média $0,58$ transações a cada segundo do teste, seguido do *VoltDB*, que executou $0,51$ transações por segundo. O *NuoDB* apresentou um tempo

intermediário, restando novamente ao *CockroachDB* a última colocação. O desvio padrão mostra que o grau de dispersão entre os resultados foi muito baixo, evidenciando uma boa uniformidade na execução.

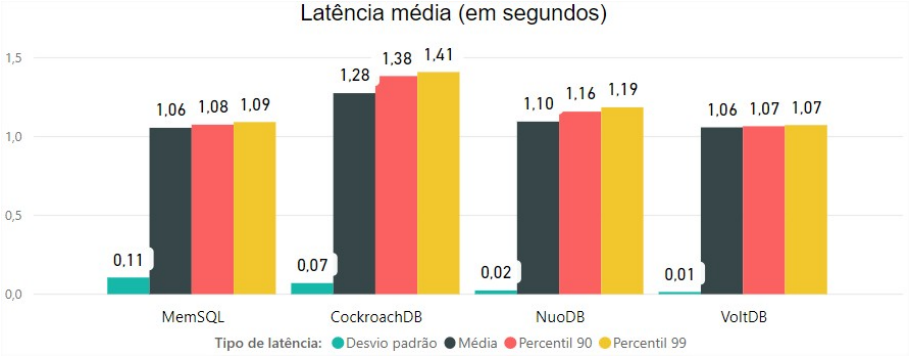


Figura 4. Médias de latência por segundo obtidas no *benchmark Voter*.

O gráfico da Figura 4 apresenta a média das latências das transações obtidas para o *Voter*. Observa-se novamente uma maior latência obtida pelo *CockroachDB*. O desvio padrão mostra que o grau de dispersão entre os resultados foi muito baixo, mostrando uma boa uniformidade na execução. O teste do *Voter* mostra uma situação muito diferente da observada com o mesmo teste realizado com o *YCSB* (Figura 2). No *Voter*, a variação obtida entre um produto e outro é pequena, tanto nas média quanto nos percentis 90 e 99. Isso demonstra que, em transações menores, como é o caso deste *benchmark*, os produtos operam de forma semelhante.

O *Voter* visa uma exploração mais acentuada das características que se esperam de um BD *NewSQL*. Como comentado anteriormente, suas transações consistem em uma votação por número de telefone em um participante do programa *American Idol*. Assim, este *benchmark* acarreta um estresse maior sobre os produtos testados, simulando diversos usuários votando em seu participante favorito. Os BDs *MemSQL* e *VoltDB* apresentaram resultados muito semelhantes, demostrando que são capazes de lidar com várias requisições simultâneas com baixa latência. O *NuoDB* apresenta uma arquitetura com níveis mais complexos de armazenamento, o que prejudica um pouco seu desempenho com um grande volume de transações mais simples. Já o *Cockroach* apresentou o pior resultado. Sua arquitetura, apesar de trazer novos algoritmos, ainda fica limitada a algumas operações que utilizam o disco, depreciando seu desempenho em relação aos demais.

6. Conclusão

Este trabalho analisa quatro produtos que se baseiam no paradigma NewSQL. A técnica de *benchmark* foi empregada, sendo os *benchmarks* escolhidos gerenciados através de um *framework* chamado *OLTP-Bench*. Os benchmarks usados apresentam características específicas para a avaliação de diferentes cenários de ambientes transacionais OLTP. O benchmark *YCSB* é amplamente utilizado para avaliação de BDs distribuídos e possui uma estrutura mais complexa (maior número de tabelas) que envolve uma combinação de transações que vão desde escritas e leituras pesadas à varreduras em tabelas. Já o *Voter*, apresenta uma estrutura mais simples e possui um foco na saturação do BD com diversas requisições rápidas (inserções e atualizações) em um pequeno conjunto de tabelas (3).

Os resultados obtidos revelaram que a solução *MemSQL* se manteve à frente nas características observadas, obtendo alta taxa de *throughput* e baixa latência. Os produtos *VoltDB* e *NuoDB* se comportaram de maneira semelhante na maioria dos resultados analisados, mesmo com as considerações sobre as restrições da versão gratuita do *NuoDB*. O SGBD *CockroachDB* apresentou os piores resultados, com discrepâncias consideráveis nas métricas observadas, principalmente nas taxas médias de transações por segundo.

Como trabalho futuro, pretende-se realizar a mesma análise utilizando particionamento geográfico dos nodos para uma verificação mais aprofundada das características analisadas. Além disso, incluir a avaliação de BDs tradicionais para as mesmas métricas, para comprovar as vantagens que o paradigma *NewSQL* evidencia.

Referências

Difallah, D. E., Pavlo, A., Curino, C., and Cudre-Mauroux, P. (2013). Oltp-bench: An extensible testbed for benchmarking relational databases. *Proc. VLDB Endow.*, 7(4).

Grolinger, K., Higashino, W. A., Tiwari, A., and Capretz, M. A. (2013). Data management in cloud environments: Nosql and newsql data stores. *JoCCASA*.

Gurevich, Y. (2015). *Comparative Survey of NoSQL/NewSQL DB Systems*. PhD thesis, The Open University.

Hajoui, O., Dehbi, R., Talea, M., and Batouta, Z. I. (2015). An advanced comparative study of the most promising nosql and newsql databases with a multi-criteria analysis method. *Journal of Theoretical & Applied Information Technology*, 81(3).

Kaur, K. and Sachdeva, M. (2017). Performance evaluation of newsql databases. In *2017 International Conference on Inventive Systems and Control (ICISC)*, pages 1–5.

Knob, R. R. (2018). Análise e benchmarking das soluções newsql cockroachdb, memsql, nuodb e voltdb. TCC, Universidade Federal de Santa Catarina.

Labs, C. (2018). Architecture overview. `https://www.cockroachlabs.com/docs/stable/architecture/overview.html#goals-of-cockroachdb` Último acesso em: 21/06/2018.

MemSQL (2018). Memsql architecture: Technology innovations power convergence of transactions and analytics. `https://www.memsql.com/content/architecture/` Último acesso em: 06/10/2018.

Oliveira, J. and Bernardino, J. (2017). Newsql databases-memsql and voltdb experimental evaluation. In *KEOD*, pages 276–281.

Pavlo, A. and Aslett, M. (2016). What's really new with newsql? *SIGMOD Rec.*, 45(2).

Stonebraker, M. (2012). Newsql: An alternative to nosql and old sql for new oltp apps. *Communications of the ACM. Retrieved*, pages 07–06.

VoltDB (2013). Using voltdb. `http://downloads.voltdb.com/documentation/UsingVoltDB.pdf` Último acesso em: 05/06/2018.

VoltDB (2015). Voltdb technical overview. `http://www.odbms.org/wp-content/uploads/2013/11/VoltDBTechnicalOverview.pdf` Último acesso em: 05/06/2018.

Unificação de Dados de Saúde Através do Uso de Blockchain e Smart Contracts

Bruno Machado Agostinho[1], Geomar André Schreiner[1], Fernanda Oliveira Gomes[1], Alex Sandro Roschildt Pinto[1], Mario Antônio Ribeiro Dantas[2]

[1]Programa de Pós-Graduação em Ciência da Computação
Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brasil

[2]Programa de Pós-Graduação em Ciência da Computação
Universidade Federal de Juiz de Fora (UFJF)
Juiz de Fora – MG – Brasil

{bruno.agostinho,schreiner.geomar,fernanda.gomes}@posgrad.ufsc.br

a.r.pinto@ufsc.br, mario.dantas@ice.ufjf.br

Abstract. *In recent years there have been several proposals aimed at centralizing and manipulating health data, such as the electronic medical records. Because this type of data are highly sensitive, issues as how to ensure data confidentiality have always been a chalenge. The emergence of technologies such as blockchains and smart contracts has brought new approaches to the manipulation of these data. This paper proposes the use of blockchain in conjunction with smart contracts for centralization and sharing of health data. Preliminary experiments demonstrated the feasibility of securely storing and retrieving data through the use of two pairs of asymmetric keys*

Resumo. *Nos últimos anos houveram diversas propostas visando a centralização e manipulação de dados de saúde, como prontuário eletrônico do cidadão. Por se tratarem de dados altamente sigiloso, problemas de como garantir a confidencialidade dos dados sempre foram um entrave. O surgimento de tecnologias como blockchains e smart contracts vem trazendo novas abordagens possíveis na manipulação desses dados. Este trabalho apresenta uma proposta de utilização de blockchain em conjunto com smart contracts para centralização e compartilhamento de dados de saúde. Experimentos preliminares demonstraram a viabilidade do armazenamento e recuperação dos dados de forma segura, através da utilização de dois pares de chaves assimétricas.*

1. Introdução

Tem-se verificado nos últimos anos uma discussão sobre as diversas formas de manusear e compartilhar dados de saúde [Kluge 2007], como por exemplo o prontuário eletrônico dos cidadãos. Estes dados devem ser acessadas em diversas esferas do atendimento por diferentes profissionais ou pelo próprio paciente. Como estas informações podem possuir diversas fontes distintas (hospitais, unidades básicas de saúde, UPAs e etc) é pertinente que hajam soluções de interoperabilidade que sejam capazes de concentrar os dados referentes ao paciente e seus respectivos atendimentos e procedimentos.

Porém, dados na área de saúde demandam uma atenção especial no que diz respeito à privacidade, pois toda informação gerada em consultas e procedimentos possui um sigilo entre o agente de saúde (médico, enfermeiro, entre outros) e paciente. Além disso, problemas como troca de informações entre médicos e quais pacientes os agentes devem ter acesso trazem ainda mais complexidade para o cenário. Recentemente, uma nova abordagem vem chamando a atenção para aplicações na área de saúde e interoperabilidade de dados, a utilização de *blockchains*.

A *blockchain* é uma tecnologia que lida com informações em aplicações *Peer-to-peer* [Nakamoto 2008]. Cada nodo pertencente a rede é uma ferramenta de armazenamento e validação dos dados. Os nodos validam as informações e entram em um consenso sobre quais dados devem ser inseridos na *blockchain*. Aliado ao crescimento da *blockchain* temos a ascensão dos *smart contracts*. Proposto pela primeira vez por *Nick Szabo* [Szabo 1997], os *smart contracts* ganharam popularidade com o lançamento da *cryptomoeda* Ethereum[1]. Eles consistem na construção de contratos que podem ser utilizados e validados por uma ou mais partes a fim de estabelecer troca de recursos de maneira segura.

Durante muito tempo diversas restrições envolvendo a manipulação de dados médicos foram consideradas entraves para a aplicações desse tipo. A ascensão do conceito de *blockchain* vem trazendo novas perspectivas e abordagens a antigos problemas de pesquisa. A unificação de dados de saúde é apenas um deles. A forma de utilização de dados proposta no contexto de *blockchain*, assim como os conceitos de *smart contracts*, traz uma nova gama de possibilidades de aplicações para utilização e troca de dados.

Sendo assim, este trabalho tem como objetivo apresentar uma nova solução para o problema de interoperabilidade de dados de saúde utilizando *blockchain* como meio de armazenamento e *smart sontracts* para o compartilhamento destas informações de maneira segura.

O restante deste artigo está organizado conforme segue. A Seção 2 apresentando alguns conceitos relacionados para o melhor entendimento da proposta. Na Seção 3 serão apresentados alguns trabalhos relacionados utilizados para comparação e validação da proposta. Já a Seção 4 apresentada a proposta de aplicação para centralização dos dados de saúde, tendo seus resultados experimentais apresentados na Seção 5. A Seção 6 apresenta as conclusões preliminares e trabalhos futuros.

2. Preliminares

Nesta Seção são apresentados os principais conceitos envolvidos no trabalho. Inicialmente é apresentada, de maneira breve, a arquitetura da *blockchain* e como esta opera. Então são apresentados os algoritmos de consenso utilizados para verificação da confiabilidade dos nodos e dos dados.

2.1. Blockchain

Segundo [Tasatanattakool and Techapanupreeda 2018], *blockchain* é uma forma de armazenamento de dados não centralizada, confiável e difícil de utilizar para fins fraudulentos. Já para [Saraf and Sabadra 2018], pode ser definida como um livro-razão distribuído, em

[1] `https://www.ethereum.org/`

uma arquitetura *peer-to-peer*, onde todos os nodos conectados possuem uma cópia dos dados, sem precisar de um banco de dados centralizado. Sendo desenvolvida em uma arquitetura distribuída,a *blockchain* pode ser considerada um sistema puramente *peer-to-peer* [Tama et al. 2017].

O funcionamento de uma *blockchain* acontece através de um conjunto de blocos conectados de maneira imutável. Caso haja uma tentativa de alteração nos dados de um bloco, todos os blocos a partir deste passam a ser inválidos. Isso ocorre pois cada bloco aponta para seu anterior através de um *HASH*. Para a geração deste *hash*, o bloco utiliza o conteúdo do bloco anterior em conjunto com o seu, gerando uma chave onde qualquer alteração pode fazer os blocos invalidarem a ligação. Uma vez que existe a premissa de que um nodo não pode confiar nos demais, os blocos são adicionados à cadeia através dos algoritmos de consenso. Estes foram projetados para que os mineradores da rede não consigam ou não tenham vantagens em manipular dados. Mineração é o processo de introduzir um novo bloco na *blockchain*. Cada nó utiliza a cadeia para verificar se a transação é legítima e se não utiliza *tokens* já gastos [Tama et al. 2017]. Algumas arquiteturas de *blockchain* podem fornecer recompensa ao nodo que inseriu o bloco na rede. Outras pagam apenas para os nodos que ajudaram a validar as transações. Essas recompensas podem ser chamadas de camada de incentivo [Yuan and Wang 2018].

Existe ainda, um tipo de específico de aplicação dentro do contexto de *blockchain* que vem ganhando destaque. Tendo sido baseados na proposta de [Szabo 1997], os *smart contracts* consistem em uma camada acima das *blockchains* convencionais. Funcionando como classes estáticas, os contratos podem ser executados por usuários para diversas funções além de transferência de ativos. Estes possuem recursos próprios para controle de acesso, invalidação do contrato, controle de saldo entre outras funcionalidades.

2.2. Algoritmos de Consenso

Para resolver o problema de falta de confiança entra os nodos de uma *blockchain*, foram desenvolvidos algoritmos de consenso para que apenas um bloco seja inserido por vez na cadeia. Segundo [Watanabe et al. 2015] um algoritmo de consenso é um conjunto de regras que permite que os usuários cheguem a um acordo mútuo. Atualmente o algoritmo mais utilizado em *blockchains* é chamado de *Proof-of-Work* (POW).

O algoritmo de consenso *POW* foi proposto por [Nakamoto 2008] como uma função de custo baseado no trabalho de [Back 2002]. Sua proposta visa gerar um esforço computacional através da geração de *HASHs* onde o *HASH* gerado seja menor que a função de custo da rede. Para isso, um número aleatório, normalmente chamado de *nonce*, tem que ser gerado por diversas tentativas e utilizado em conjunto com os dados do conteúdo do bloco atual e do *HASH* do bloco anterior. O sistema redimensiona a função de custo para que cada bloco da rede proposta (*Bitcoin*) seja inserido a cada 10 minutos aproximadamente. Essa abordagem também evita problemas de nodos mal intencionados, pois dificilmente o mesmo nodo conseguirá inserir dois blocos simultâneos na rede, tendo um bloco malicioso desconsiderado em alguma tentativa de manipulação.

Devido ao alto custo computacional requerido para utilização do protocolo *POW*, alternativas tem sido propostas visando a diminuição do uso de recursos. A alternativa ao *POW* mais utilizada atualmente é o protocolo *Proof-of-Stake*, que se baseia na ideia

de aplicar "um voto por unidade de participação no sistema" na escolha do nodo que vai inserir o próximo bloco, onde a participação pode ser medida pela quantidade de unidades (*tokens*, criptomoedas) pertencentes a um nodo específico [Bentov 2016]. Dessa maneira, nodos que possuem mais *tokens* tendem a ter preferência na inserção de novos blocos.

3. Trabalhos Relacionados

Do ponto de vista acadêmico, a utilização de dados médicos para troca de informações dentro da internet não pode ser considerado algo recente. Embora sempre tenham existidos obstáculos, na maioria das vezes devido ao sigilo dos dados, diversas propostas têm sido desenvolvidas com o passar dos anos. *Chen*, em [Chen et al. 2012] propôs a utilização de uma integração de *clouds* públicas e privadas para troca de informações de prontuários eletrônicos. Em [Sucurovic 2007] foi detalhado o sistema *MEDIS* para centralização de dados de saúde, assim como as abordagens de segurança no acesso ao sistema. Em um sistema utilizando *blockchain*, [Azaria et al. 2016] propôs a utilização de contratos para mapeamento de dados, permissões e transição de estados, em uma *blockchain* que funciona como um ponteiro para bancos de dados descentralizados. *Yue*, em [Yue et al. 2016] propôs o armazenamento dos dados médicos em uma *blockchain*, e o desenvolvimento de *gateways* utilizados por usuários para o acesso a troca de informações.

O trabalho proposto por [Sucurovic 2007] teve um foco na junção de dados de sistemas de diversas instâncias de unidades de saúde. Visando a segurança, o trabalho foi voltado para políticas de acesso aos dados e ao sistema. Embora seja parte importante, não foi mencionado nenhuma prática que impedisse a manipulação dos dados uma vez que uma pessoa consiga ter acesso ao banco de dados por meio de ataques.

No trabalho de [Chen et al. 2012], foi criada uma integração entre *clouds* públicas e privadas para troca de informações. Embora tenha sido pensada em uma estrutura onde os dados ficam armazenados de maneira sigilosa, esta proposta apresentou um primeiro ponto de vulnerabilidade ao inserir uma forma de acessar as informações de um paciente como válvula de escape para emergências. Embora existam algumas regras para que isso aconteça, isso pode vir a se tornar o foco de atacantes para ter acesso a informações sigilosas. Outro ponto crítico é o acesso ao banco de dados. Uma vez que um atacante consiga acessar uma das estruturas em nuvem ele pode conseguir manipular os dados mesmo sem conseguir ler os mesmo.

Em [Azaria et al. 2016], foi proposta a utilização de *blockchains* em conjunto com *smart contracts* para gerenciamento de acessos e ponteiros para os dados médicos. Assim como os trabalhos citados anteriormente, no caso de um acesso direto a um dos servidores de BD os dados poderiam ser manipulados, estando cifrados ou não.

O trabalho de [Yue et al. 2016] apresenta muitas semelhanças com esta proposta. Em uma estrutura que parece garantir disponibilidade, confidencialidade, autenticidade e integridade dos dados, os autores propuseram o armazenamento de dados em *blockchain* e o desenvolvimento de *gateways* para leitura e troca de dados, utilizando chaves para cifrar e decifrar os dados. Embora existam semelhanças, os autores deixaram a desejar nas especificações da estrutura de troca de dados. Após sugerir a utilização de uma tabela única para inserir os dados compartilhados, as poucas informações do desenvolvimento não deixam realmente claro como o sistema faz a manipulação dos dados.

4. Proposta

Para o desenvolvimento da proposta de centralização de dados de saúde utilizando *blockchain*, foram utilizadas como modelo as camadas propostas no trabalho de [Yuan and Wang 2018]. Neste trabalho, os autores propuseram uma formalização no desenvolvimento de *blockchains*, dividindo e esclarecendo o funcionamento de cada uma das camadas, como pode ser visto na Figura 1. Embora tenham sido propostas 6 diferentes camadas (Dados, Rede, Consenso, Incentivo, Contrato e Aplicação), a esta proposta utilizou apenas 4, deixando de fora as camadas de Incentivo e Consenso. A camada de Consenso deve ser especificada e desenvolvida posteriormente enquanto a camada de Incentivo deve ser analisada para ver sua pode ser adequada a este tipo de rede.

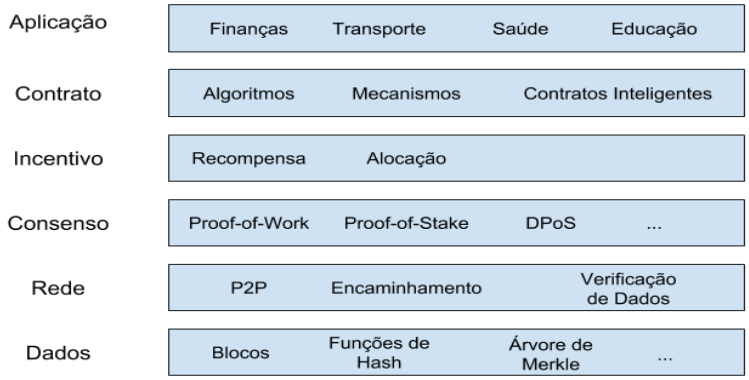


Figura 1. Camadas propostas por [Yuan and Wang 2018].

4.1. Camada de Dados

Nesta camada ficarão armazenados todos os dados gerados por qualquer tipo de interação entra pacientes, médicos, enfermeiros, procedimentos e até aparelhos hospitalares. Essas interações serão tratadas como transações, sendo armazenadas inicialmente com o status de não confirmadas. Como se tratam de dados sensíveis, o conteúdo armazenado de todas as transações é cifrado através de criptografia assimétrica, podendo serem lidas apenas pela entidade originadora da transação.

Sempre que uma interação entre paciente e agentes de saúdes ocorrer, os resultados devem ser inseridos como transações. Cada interação pode gerar até 3 transações. A primeira delas tendo como origem o paciente, a segunda o agente de saúde e a terceira para uma base de análise de dados pública. Na primeira e segunda transação, os conteúdos são cifrados pelas chaves das entidades originadoras antes de serem inseridos. A terceira transação tem como objetivo publicar dados para análise de maneira pública e é gerada utilizando apenas dados que não sejam sensíveis. Não deve ser possível identificar o paciente ou médico relacionado a essa interação. A Figura 2 demonstra como será estruturada os dados dentro da *blockchain*. Cada bloco pode alocar até N transações, variando de acordo com o tamanho máximo configurado para os blocos. Uma transação consiste nos dados de uma interação, que permanecem cifrados, e em uma assinatura digital, que visa garantir a integridade dos dados inseridos.

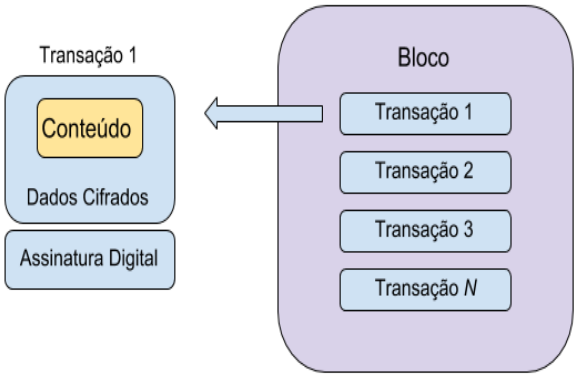


Figura 2. Bloco e Transações.

4.2. Camada de Rede

Uma vez que esta proposta pode ser aplicada para dados de saúde do país inteiro e que deve ser realizada de maneira a manter não só o sigilo, mas também evitando a propagação de falsas informações, esta deve possuir permissões. Apenas alguns nodos pertencentes à rede devem poder inserir os blocos e transações. Assim sendo, os tipos de integrantes dessa camada foram divididos em quatro perfis:

4.2.1. Usuários

O perfil de usuário na rede será utilizado pelos pacientes e agentes de saúde. Cada usuário será cadastrado em uma *blockchain* secundária tendo um identificador único (CPF) vinculado á uma chave pública que será utilizada para assinar as transações. O usuário também deverá portar um dispositivo inteligente (*token*, *smart card*) contendo duas chaves. A primeira é a chave privada correspondente à chave pública que está localizada na *blockchain* secundária. A segunda é uma chave pública que será utilizada para cifrar os dados referentes às interações entre usuários. O usuário ainda terá guardada a respectiva chave privada que pode ser utilizada para decifrar estes dados. O fluxo de uma interação entre usuários acontece da seguinte maneira:

1. Os dados da interação são gerados por um dispositivo da entidade de saúde.
2. O usuário utiliza o dispositivo inteligente.
3. Os dados da interação são cifrados utilizando a chave pública do dispositivo.
4. O dispositivo gera um *hash* dos dados cifrados e utilizada a chave privada do dispositivo para cifrar o *hash*
5. O dispositivo inteligente devolve os dados no formato de transação.
6. A transação é enviada para a lista de transações não confirmadas

Esse fluxo será repetido para cada usuário pertencente a interação, mantendo cópias da transação com acesso restrito a cada um.

4.2.2. Entidades de Saúde

As entidades de saúde serão os responsáveis pela inserção de transações não confirmadas na *blockchain*. Uma entidade de saúde pode ser um hospital, clínica, uma ambulância ou

qualquer outra entidade que esteja apta a prestar atendimento a um cidadão. Possuindo um identificador único para a entidade, deve ser possível a configuração de diversos dispositivos pertencentes a ela. Todos eles devem realizar as transações como se fossem um só, utilizando o mesmo identificador.

As entidades também são responsáveis pela geração das transações com dados da interação que não identificam o cidadão. Cada tipo de interação permitida deve ter um modelo de dados previamente cadastrado onde ficam marcados quais deles são confidenciais e quais podem ser enviados de maneira aberta.

4.2.3. Supervisor de Rede

Os nodos supervisores de rede ficam responsáveis pelas confirmações de transações e inserção de novos blocos na cadeia. Por ser uma *blockchain* do tipo permissionada, apenas nodos confiáveis, e normalmente pré-selecionados, são autorizados. O número mínimo de nodos de redes disponíveis para o funcionamento da proposta é dois. Isso acontece pois um nodo sempre ficará aguardando um número mínimo de transações confirmadas para geração de um novo bloco enquanto os outros ficam responsáveis pelas confirmações.

4.2.4. Armazenamento

O armazenamento dos dados de transações e blocos da cadeia é realizado em nodos específicos para isso. Assim como os supervisores de rede, a camada de armazenamento fica sob responsabilidade dos administradores *blockchain*. Por se tratar de uma grande quantidade de dados em um cenário onde todos os nodos possuem uma cópia exata da cadeia, os dispositivos utilizados para o armazenamento necessitam de uma configuração apropriada, não contendo riscos de limitação por espaço.

4.3. Camadas de Contrato e Aplicação

A camada de contrato será utilizada como um sistema de troca de informações entre usuários da camada de rede. Uma vez que médicos pode ter a necessidade de trocar prontuários de pacientes ou até mesmo um paciente que deseje um segundo parecer médico, o sistema deve permitir que tais dados sejam manuseados de maneira sigilosa.

O sistema para troca de informações será desenvolvido como uma aplicação descentralizada, que ficará hospedada nos nodos de dados. Os usuários terão livre acesso a seus dados, que continuarão cifrados a menos que seja utilizada a chave privada.

Quando um usuário (médico, enfermeiro, paciente, etc...) desejar compartilhar determinados dados, este deverá escolher qual o usuário vai receber os dados. Após entrar com a chave privada para decifrar os dados a serem enviados, o sistema criará um novo par de chaves, cifrando a informação com a chave pública criada e assinando com a chave pública do usuário remetente. Essa informação ficará contida em um *smart contract* que só poderá ser acessado pelo usuário remetente e pelos usuários destinatários. Ainda assim, para dificultar qualquer acesso indesejado, a chave privada para decifrar os dados será enviada em outro contrato, este será cifrado com a chave pública do destinatário. No caso de mais de um destinatário, múltiplos contratos com a mesma chave serão criados.

Após a criação dos contratos, os usuários destinatários passam a ter acesso aos dados de ambos os contratos pelo sistema decentralizado. A intenção em separar os dados em contratos diferentes é de invalidar o uso de qualquer um dos dois de maneira isolada.

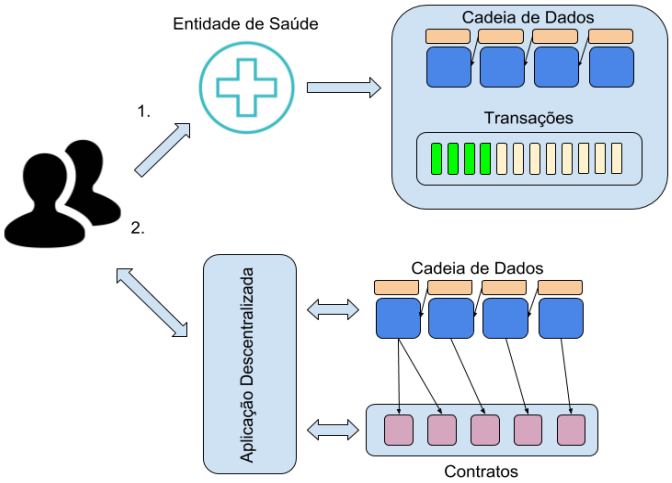


Figura 3. Arquitetura da Proposta.

A Figura 3 mostra uma visão geral sobre os possíveis fluxos de utilização de dados dentro da *blockchain* proposta. A primeira possibilidade mostra a iteração entre usuários gerando dados para uma entidade de saúde e esta inserindo novas transações na rede da *blockchain*. As transações vão sendo confirmadas e inseridas nos blocos que acabam na cadeia. No segundo fluxo é mostrada a relação entre usuários, sistema, *blockchain* e contratos.

5. Experimentos Preliminares

Para os primeiros testes de validação da proposta, foi desenvolvido um ambiente de simulação de uma *blockchain*. Os nodos da *blockchain* foram desenvolvidos em *Node.js* utilizando instâncias da biblioteca *Express*. A comunicação foi realizada através de serviços, simulando uma rede *peer-to-peer*. Cada nodo contém os serviços para listagem dos nodos da cadeia, inserção de novo nodo, verificar integridade, inserir, retornar dados e decifrar uma transação. Foram utilizados 4 instâncias para os testes. Para armazenamento, foi utilizado o Mongo DB. O objetivo dos primeiros testes foi validar a ideia de utilização de dois pares de chaves assimétricas para manipulação e garantia de confidencialidade dos dados. Para isso, cada instância foi criada com dois pares de chaves *RSA*, utilizando a biblioteca URSA. No BD, foram criadas duas coleções. A primeira foi utilizada para vincular uma chave pública a um identificador do usuário, que conforme a proposta utilizará o CPF. A Figura 4 mostra como ficaram armazenadas as chaves públicas no Mongo DB.

Para a validação do uso dos pares de chaves, foi desenvolvido um teste para inserção dos dados cifrados e assinados no BD. Antes de enviar os dados para os outros nodos, o conteúdo da transação de teste foi definido como “Esta transação não pode ser acessada.”. O conteúdo foi então cifrado pela chave pública do par 1. Para garantir a integridade, um *hash SHA1* foi calculado em cima do conteúdo cifrado e este *hash* foi cifrado utilizando a chave privada do par 2. A transação então foi enviada para o BD contendo as informações: ID, CPF, DADOS e ASSINATURA.

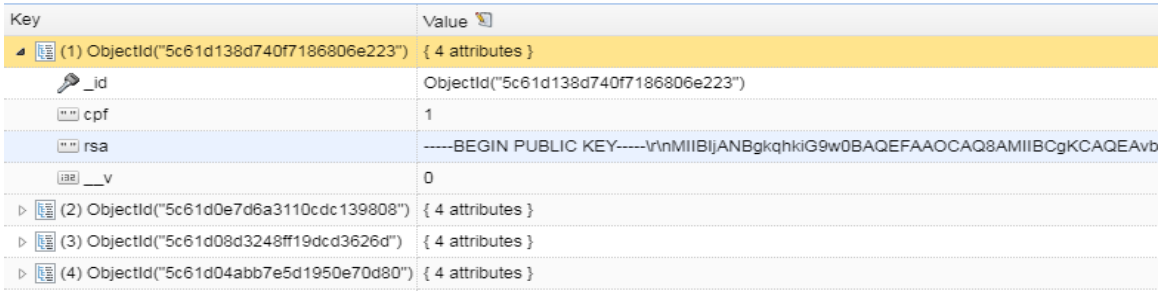


Figura 4. Armazenamento das chaves públicas.

Figura 5. Interface de validação.

Para o acesso às transações foram desenvolvidas algumas telas utilizando *HTML* para simular a parte correspondente a aplicação decentralizada da proposta. Utilizando o identificador o usuário pode baixar os dados da transação e decifra-los utilizando sua chave privada do par 1.

A Figura 5 apresenta um exemplo de utilização da funcionalidade de acesso aos dados. Na Figura 5 (A) foi utilizado o identificador da transação para baixar os dados da transação ainda cifrado. A Figura 5 (B) mostra os dados da transação após a utilização da chave privada. Conforme mencionado anteriormente, a transação de teste continha uma *string* de valor ”Esta transação não pode ser acessada”.

6. Conclusão e Trabalhos Futuros

O trabalho de pesquisa apresentado neste artigo sugere a utilização de *blockchains* em conjunto com *smart contracts* a fim de viabilizar a centralização e manipulação de dados médicos. Foram realizados experimentos iniciais em relação a validação da proposta de utilizar dois pares de chaves assimétricas para garantir a confidencialidade dos dados. Os resultados preliminares demonstraram a viabilidade o uso de pares e da utilização do formato de transação sugerido, com os dados cifrados por um par de chaves e assinado por outro.

Por se tratar de um trabalho em andamento, é necessária a implementação completa da proposta para a devida validação. São necessários testes utilizando uma *blockchain* real e não mais simulada. Uma hipótese a fim de otimizar o armazenamento é usar a *blockchain* apenas os dados de *hash* das transações mantendo os dados efetivos no *MongoDB* .

Referências

Azaria, A., Ekblaw, A., Vieira, T., and Lippman, A. (2016). Medrec: Using blockchain for medical data access and permission management. In *2016 2nd International Conference on Open and Big Data (OBD)*, pages 25–30.

Back, A. (2002). Hashcash - a denial of service counter-measure. http://www.hashcash.org/papers/hashcash.pdf. Acessado: 11/02/2019.

Bentov, Iddo; Pass, R. S. E. (2016). Snow white: Provably secure proofs of stake. https://eprint.iacr.org/2016/919.pdf. Acessado: 11/02/2019.

Chen, Y.-Y., Lu, J.-C., and Jan, J.-K. (2012). A secure ehr system based on hybrid clouds. *Journal of Medical Systems*, 36(5):3375–3384.

Kluge, E.-H. W. (2007). Secure e-health: Managing risks to patient health data. *International Journal of Medical Informatics*, 76(5):402 – 406.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf. Acessado: 11/02/2019.

Saraf, C. and Sabadra, S. (2018). Blockchain platforms: A compendium. In *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, pages 1–6.

Sucurovic, S. (2007). Implementing security in a distributed web-based ehcr. *International Journal of Medical Informatics*, 76(5):491 – 496.

Szabo, N. (1997). Formalizing and securing relationships on public networks. *First Monday*, 2(9).

Tama, B. A., Kweka, B. J., Park, Y., and Rhee, K. (2017). A critical review of blockchain and its current applications. In *2017 International Conference on Electrical Engineering and Computer Science (ICECOS)*, pages 109–113.

Tasatanattakool, P. and Techapanupreeda, C. (2018). Blockchain: Challenges and applications. In *2018 International Conference on Information Networking (ICOIN)*, pages 473–475.

Watanabe, H., Fujimura, S., Nakadaira, A., Miyazaki, Y., Akutsu, A., and Kishigami, J. J. (2015). Blockchain contract: A complete consensus using blockchain. In *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*, pages 577–578.

Yuan, Y. and Wang, F. (2018). Blockchain and cryptocurrencies: Model, techniques, and applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(9):1421–1428.

Yue, X., Wang, H., Jin, D., Li, M., and Jiang, W. (2016). Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control. *Journal of medical systems*, 40:218.

Um *Data Warehouse* Textual em Língua Portuguesa: Estudo de caso do sentimento dos usuários do Twitter durante a eleição de 2018

Instituto Federal de Educação, Ciência e Tecnologia Catarinense – Campus Camboriú
Caixa Postal 2016 – 88.340-055 – Camboriú – SC – Brasil

Jonathan Vinícius Suter, Rodrigo Ramos Nogueira,Tatiana Tozzi, Daniel Fernando Anderle, Rafael de Moura Speroni

{jonathan.vinicius.suter, wrkrodrigo, tatitozitt}@gmail.com, {daniel.anderle, rafael.speroni}@ifc.edu.br

Resumo. *As redes sociais cada dia mais causam impacto no cotidiano das pessoas e organizações, neste contexto, o Twitter, no qual um usuário escreve uma expressão com até 280 caracteres e outras pessoas podem ver ou compartilhar novamente essa mesma expressão ou a sua própria. Este artigo apresenta um trabalho que tem como objetivo consumir o grande repositório de dados que é o Twitter, e, a partir dele criar um Data Warehouse no qual é possível analisar os textos, as expressões contidas. Nesta proposta, são incluídos métodos de pré-processamento dos textos. Também para enriquecer essa base com a análise de sentimento, além do projeto do banco de dados a proposta inclui um método classificador para os textos, utilizando aprendizado de máquina, que é capaz de predizer um sentimento relacionado a um Tweet, seja ele positivo, neutro ou negativo.*

Abstract. *Social networks are increasingly impacting everyday people and organizations, in this context Twitter, in which a user writes an expression with up to 280 characters and other people can see or share that phrase again or their own. This paper presents a work that aims to consume the great data repository that is Twitter, and from it to create a Data Warehouse in which it is possible to analyze the texts, the expressions contained. In this proposal, methods of preprocessing texts are included. Also to enrich this base with the analysis of feeling, in addition to the project of the database the proposal includes a classifier method for the texts, using machine learning, that is able to predict a feeling related to a Tweet, be it positiva, neutral or negative.*

1. Introdução

Desde o início da Web, o volume de dados que estão nos repositórios na rede mundial tem crescido de forma exponencial, atualmente são cerca de 200 milhões de sites ativos na Internet1, dos quais, apenas a rede social Twitter gera, em média, 500 milhões de postagens por dia. Tal explosão de dados, levou a um estudo do IDC (Institute Data Corporation) que estima que até 2020 serão gerados 44 zettabytes de dados em todo mundo. No entanto, com o crescente aumento de dados de maneira escalável fazem com que os métodos tradicionais de exploração desses dados têm se tornado inadequados,

segundo (CAMILO; SILVA, 2009). Desta forma, torna-se necessário não apenas a revisão dos métodos atuais, mas principalmente a criação de novos métodos de exploração, mais rápidos e precisos.

Com o crescimento da utilização das redes sociais, os conteúdos compartilhados demonstram características associadas ao perfil de cada usuário, principalmente seus interesses e opiniões sobre os mais diversos assuntos. No contexto da expressão de pensamentos e compartilhamento desses sentimentos, as redes sociais são repositórios gigantes de dados a respeito dos mais variados assuntos, objetos, pessoas, comportamentos. No caso do Twitter, o poder de se definir um raciocínio em poucos caracteres a torna singular neste aspecto, sendo um facilitador da dispersão de idéias.

Devido ao volume de dados, torna-se humanamente impossível analisar as expressões de ideias, sendo necessário o desenvolvimento de um mecanismo que seja capaz de captar esses dados, analisar e indicar quais os sentimentos, emoções estão sendo transmitidos pelas pessoas através dos Tweets.

Para JUNQUEIRA (2018), entre diversas aplicações em um conjunto linguístico baseado em textos do Twitter, se destacam as pesquisas que exploram a análise de sentimento. O processo de análise de sentimentos consiste na abordagem computacional que, com a utilização de técnicas de processamento de linguagem natural e aprendizagem de máquina, tem o objetivo de julgar textos a fim de determinar sentimentos e opiniões presentes em frases. Análise de sentimentos também é comumente conhecida por vários outros termos, tais como: extração de opinião, mineração sentimento, análise de subjetividade, análise afetiva, análise de emoções e mineração de opinião.

Em redes sociais, a análise de sentimentos é utilizada para verificar a polaridade de opiniões e pensamentos dos usuários, ou seja, se as opiniões e pensamentos são positivos ou negativos. Assim, a análise de sentimentos se tornou campo de interesse de vários setores, funcionando como ferramenta de *feedback* sobre o que as pessoas pensam, segundo (CAVALCANTE, 2017).

A análise multidimensional da rede social pela perspectiva do sentimento pode ser útil em diversos contextos desde marcas avaliando seu produto,até mesmo como no caso do objeto de estudo, avaliar o cenário político. Deste modo, este artigo apresenta as etapas da criação de um Data Warehouse alimentado com dados da rede social Twitter e efetuar o enriquecimento semântico partir do sentimento dos dados extraídos.

2. Trabalhos Relacionados

A análise de sentimentos é uma sub-área da inteligência artificial em ascensão tendo diversas aplicações, principalmente no marketing de produtos e político. Sabendo da ampla utilização do Twitter para armazenar dados e expressar sentimentos, o mesmo tem sido amplamente empregado como fonte de caso de estudo para diversos trabalhos nesta área. JUNQUEIRA (2018), realizou a coleta de 988.512 textos do Twitter, os rótulos foram inseridos manualmente, posteriormente foram avaliados os métodos de aprendizado de máquina, onde o melhor método foi o SVM com uma acurácia de 95,7%.

Um estudo de ARAÚJO, Mateus et. al (2016) sobre o funcionamento dos métodos de análise de sentimento no contexto das redes sociais. Utilizando duas bases com dados de redes sociais, foi feita comparação do funcionamento entre oito métodos de classificação de sentimentos e quais os resultados da análise. Utilizando a mineração de dados no Twitter, CORREA (2017) fez a extração e análise dos sentimentos dos filmes indicados ao Oscar de 2017 utilizando o algoritmo de classificação naive Bayes, baseado no teorema de Thomas Bayes, classificando os Tweets em relação ao seu conteúdo como positivo, negativo e/ou neutro. Efetuando a extração de Tweets relacionados a três categorias de notícias, NASCIMENTO, Paula et al. (2012) efetuaram a análise de sentimento, se o sentimento em relação a elas era positivo ou negativo. Utilizando classificadores, com aprendizado supervisionado, mediu-se qual era mais eficaz para este tipo de tarefa, para textos em português, especificamente.

LOCHTER et. al (2014) identificaram a necessidade de qualificadores de textos mais eficazes para auxiliar na medição da polaridade dos mesmos, dada a grande quantidade de abreviações, gírias e símbolos utilizados nas redes sociais. Para isto, utilizou-se dicionários semânticos e ontologias para auxiliar na elaboração de um comitê de classificadores que detectam automaticamente os métodos de classificação de linguagem natural mais eficazes nessa tarefa. Por sua vez, MORAES et. al. (2015) coletaram Tweets em português que foram postados durante a partida entre Brasil e Alemanha na Copa do Mundo FIFA de 2014 para identificar a polaridade. A classificação dos Tweets foi feita a mão e após, foram mostrados os resultados, a quantidade de Tweets positivos, negativos e/ou neutros.

AGUIAR et al. (2018), mediram a capacidade de um comitê de algoritmos de aprendizado de máquina para análise de sentimento em redes sociais com a língua portuguesa, usando como estudo de caso a rede social Twitter. Concluiu-se que em alguns casos, os outros algoritmos e o comitê obtiveram desempenho equivalente na mesma tarefa. TAVARES et al. (2017) apresentam uma solução de Business Intelligence para facilitar a extração de informações da rede social Twitter por organizações, efetuando a etapa de ETL, extraindo informações através do reconhecimento de entidades nomeadas, a descoberta de conhecimento em texto , inserindo os dados em uma nova base e permitindo a análise gráfica dos dados.

Para minerar dados da rede social Twitter, TREVISAM (2015) desenvolveu uma ferramenta para recuperação inteligente de dados para que pudesse efetuar a sumarização e posterior análise dos dados para que se possa extrair informações a partir desta base dados, a respeito de algum evento no mundo real.

3. Metodologia

Para PIZZANI et al. (2012), uma pesquisa bibliográfica tem vários fins, para aprimorar-se o conhecimento a respeito do assunto abordado e as tecnologias relacionadas, também para auxiliar na definição do escopo do que será desenvolvido. Por isso a primeira etapa desta pesquisa foi dedicada ao levantamento bibliográfico para se obter a fundamentação teórica sobre o que é Data Warehouse, o que é a análise de sentimento, os métodos de classificação por aprendizado de máquina, bem como os trabalhos já desenvolvidos na mesma linha de estudo (estado da arte).

Esta pesquisa também se enquadra como pesquisa tecnológica de acordo com JUNIOR et al. (2014), pois o produto final é conjunto de arquitetura, software, complementado de um conjunto de dados. Para o desenvolvimento desta etapa foi realizado a extração dos dados, mediante ao emprego de um Web Crawler, que busca os Tweets através da API disponibilizada pelo próprio Twitter. Então, os Tweets são classificados de acordo com seu conteúdo, se eles têm conteúdo positivo, neutro ou negativo relativo ao assunto.

A condução do desenvolvimento foi realizado tendo como base a arquitetura criada com base na arquitetura de um Data Warehouse de KIMBAL (2011). A Figura 1 mostra a arquitetura proposta por esta aplicação de Data Warehouse. Inicialmente efetuada a coleta dos textos assim como o pré processamento, compondo a etapa de ETL. Finalmente, após os dados pré-processados e limpos podem ser realizadas consultas OLAP para explorar o cubo de dados. As etapas da arquitetura são descritas em maior nível de detalhamento na sequência.

Figura 1. Arquitetura utilizada para coleta e Data Warehousing

Acoplado à etapa de extração da ETL, a coleta é feita por um *Web Crawler*, desenvolvido utilizando a linguagem de programação *Python*, na versão 3.6. As requisições ocorrem através do uso da biblioteca "TwitterSearch 1.0.210". Uma vez optando-se por coletar textos em língua portuguesa, sobre as eleições de 2018.

A etapa da limpeza de dados é essencial para o armazenamento de textos, pois é nela que são removidos os dados desnecessários, que além de ocupar espaço em disco, podem atrapalhar o desempenho dos métodos computacionais que utilizam os dados armazenados. Na etapa de limpeza desenvolvida durante a arquitetura Data Warehouse deste projeto foram considerados os seguintes fatores que foram removidos dos textos coletados:

a) Existência de imagens, bitmaps, gifs e etc. no meio dos textos, sendo necessária a retirada dos mesmos para que possam ser inseridos na base.

b) Retweets: devido às limitações que a API do Twitter impõe, não há como desconsiderá-los entre as requisições, diminuindo a variação dos textos e criando a necessidade de tratar os textos com essa marcação.

c) Links no meio dos Tweets. Exemplo:

d) Sequência de caracteres que são reconhecidos como de "escape" pelo compilador ou que prejudicam a construção do SQL para inserção do texto na base (sequências do tipo "\n" e aspas simples no meio da cadeia de texto)

e) Espaços vazios em "excesso". Exemplo:

f) Remoção de Stop-words.

Com os textos já limpos, seleciona-se a data do registro e é efetuada sua formatação para que possa ser inserida na base. A partir disso, os dados do Tweet estão preparados para que o mesmo possa "quebrado" e se efetue a Bag of Words. Com os dados do Tweet, as palavras são quebradas pelo script e inseridas na base de dados multidimensional. Caso a palavra já exista na base, é apenas atualizada sua frequência.

E assim, tem se um documento com os termos e sua frequência em cada Tweet e com uma consulta, sua frequência na base como um todo.

O banco de dados multidimensional armazena os textos dos *Tweets* que foram padronizados e limpos. O modelo multidimensional descrito na Figura 2 representa os dados armazenados neste artigo. No qual a tabela fato são os textos curtos (tweets) que são analisados por suas dimensões (sentimento, tempo, palavra).

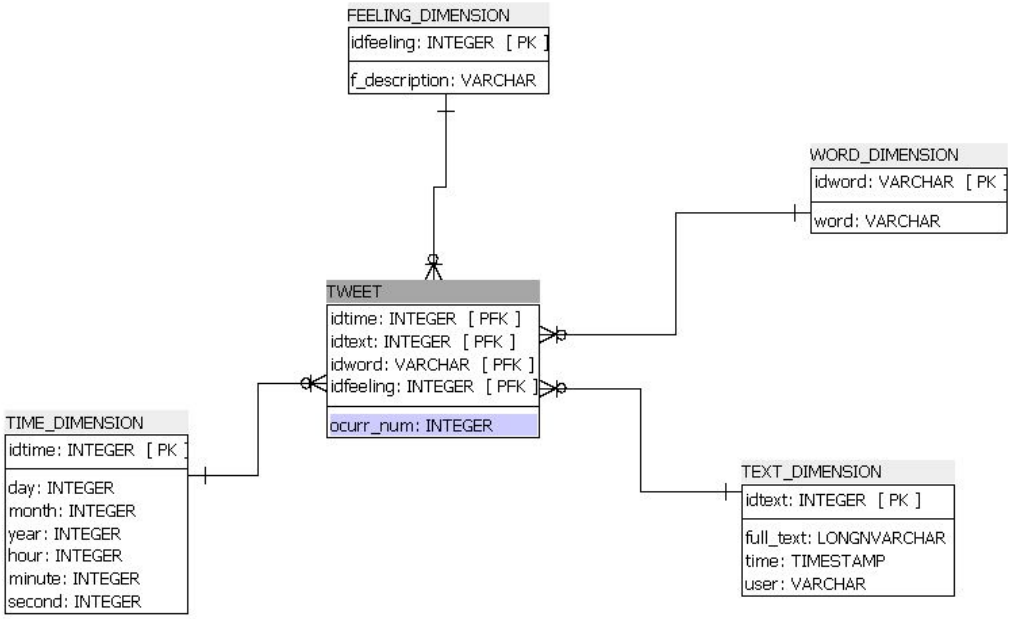


Figura 2. Modelo Multidimensional de textos e sentimentos

Foram coletados 108893 Tweets entre os meses de julho e outubro, referentes à hashtag "eleicoes2018". Após as etapas de coleta, preparação dos textos e enriquecimento semântico e, ao efetuar o treinamento do algoritmo de classificação, usando o conjunto de dados para treinamento com 1300 tweets classificados manualmente.

4. Resultados e Discussões

O ambiente desenvolvido, que visa essa integração por meio da proposta de uma arquitetura de Data Warehouse, bem como os materiais e métodos utilizados em seu desenvolvimento.

O objetivo do ambiente desenvolvido é fornecer um conjunto de dados consistentes e limpos, na forma de um conjunto de dados em um modelo multidimensional de texto do Twitter rotulados com sentimentos, de tal maneira, que possam ser consumidos aplicações externas e usuários. Sendo assim, o ambiente foi desenvolvido baseado em uma arquitetura que visa proporcionar:

- um modelo multidimensional que armazene o conjunto de textos, sentimentos e a característica temporal dos Tweets com dados em em tempo real;
- anotações semânticas de maneira dinâmica no ambiente;
- a exploração de qualquer cubo de dados de consultas multidimensionais requisitadas por aplicações e usuários.

A arquitetura proposta por esta aplicação foi desenvolvida com em um processo de ETL denominado *ETQ (Extract, Transform, Query*), que realiza consultas dinâmicas em variadas fontes de dados (principalmente dados oriundos da Web, etc.), gerando um painel visual para atender às demandas dos usuários e principalmente uma *API (Application Programing Interface)* para responder às demandas online de aplicações. Assim, os resultados do processamento *OLAP* podem integrar dados de todas as fontes relevantes às consultas realizadas.

Então, após o teste, foram qualificados os demais tweets da base e assim, explorando as dimensões do *Data Warehouse,* pode-se obter os resultados de palavras e ocorrências mostrados pelo Quadro 1.

Palavra	Identificador	Quantidade
eleições2018	93	51458
bolsonaro	80	24424
candidato	3	10559
haddad	79	9726
diz	97	8184
presidente	230	7188
contra	93	7160
eleições	341	7125
sobre	39	6443

Quadro 1. As dez palavras com maior número de ocorrências

Pode-se observar que naturalmente, o termo usado para a pesquisa dos Tweets é o que tem mais ocorrências, este pode ser desconsiderado no momento. Entretanto, a segunda palavra mais citada entre os textos é “bolsonaro”. O segundo termo mais citado é “candidato” e o terceiro é “haddad”, indicando primariamente que estes foram os candidatos mais citados.

Tendo como objetivo fazer uma análise mais objetiva sobre as eleições, foram selecionados os nomes dos candidatos e consultados os mesmos. O Quadro 2 ilustra o resultado de menções por candidato.

Código identificador	Candidato	Quantidade de citações
80	Bolsonaro	24424
79	Haddad	9726
333	Ciro	4510
545	Alckmin	1897
2524	Amoêdo	192
4640	Daciolo	1819
889	Meirelles	588
1052	Marina	1817
7067	Alvaro	139
2487	Boulos	1098
2979	Vera	61
2534	Eymael	13
13512	Goulart	78
Total	-	46362

Quadro 2. Quantidade de menções diretas por candidato

Como é possível ver no Quadro 2, entre o total de Tweets coletados, houveram 46362 com citações a candidatos à presidência. O candidato mais citado entre os Tweets foi Jair Bolsonaro, com 24424 citações em Tweets, cerca 52,68% do total de citações; Em contrapartida, o candidato com menos citações na base é o José Maria Eymael, com apenas 13: cerca de 0,028% do total de citações. Esta consulta pode ser utilizada como uma base para uma análise de repercussão de cada candidato. Demonstra numericamente quais candidatos estiveram mais à vista dos eleitores.

A análise multidimensional também permite a associação entre dimensões de um Data Warehouse. O Quadro 3 mostra a nome dos candidatos e as menções feitas à eles em relação aos sentimentos.

Candidato	Ruim	Neutro	Bom
Bolsonaro	7946	14796	1279
Haddad	3255	5681	465
Ciro	1826	1936	632
Alckmin	1163	618	81
Amoêdo	117	46	27
Daciolo	401	868	314
Meirelles	227	312	40
Marina	763	897	149
Alvaro	41	72	26
Boulos	590	251	249
Vera	14	46	1
Eymael	6	6	1
Goulart	23	55	0
Total	16372	25584	3264

Quadro 3. Quantidade de menções por sentimento

A primeira análise a ser feita é que, os candidatos que estavam à frente do pleito primeiro receberam um grande volume de tweets negativos e positivos. Sendo que do

mesmo modo, é possível observar que nenhum candidato obteve mais citações boas que ruins. Este dado reflete a polarização e o ódio muitas vezes noticiado durante a campanha[1], sendo que o sentimento geral entre os tweets foi ruim, e que nenhum candidato conseguiu obter uma grande aprovação dos eleitores, comprovado pelo grande número de abstenções na eleição de 2018. (refletindo de certa forma, muito bem como se encerrou a disputa eleitoral).

O melhor resultado, não era o esperado no início da pesquisa, partiu justamente da ocorrência de termos e sua exploração multidimensional. Um vez que o quando ordenamos os candidatos pelo seu número de citações na rede social, o ranking é muito próximo do resultado das eleições em primeiro turno O Gráfico 1 elucida tais resultados, no qual quando comparados com dados do TSE (2018) a única diferença é que os candidatos Guilherme Boulos e Marina Silva obtiveram mais citações do que votos.

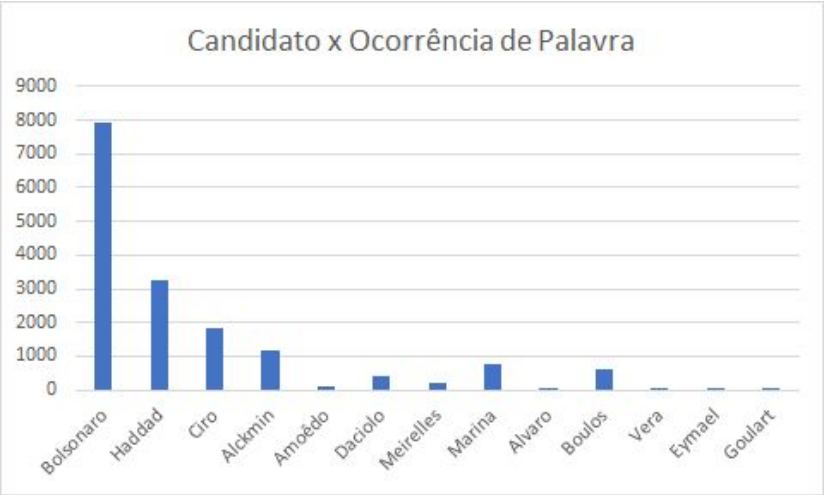


Gráfico 1. Menções por candidato no primeiro turno

6. Considerações finais

O tratamento e análise de textos escritos por pessoas, que possuem pouca ou nenhuma revisão, ainda mais em um espaço de informalidade como o Twitter, podem trazer desafios, tanto com os dados em si quanto com o sentido que eles possuem. Assim, a inserção de uma etapa para classificação dos textos como parte da ETL se tornou essencial para automatizar essa tarefa, que pode ser bastante morosa para um humano. Desta forma, o pré-processamento dos textos para que os mesmos possam entrar na base de dados já limpos e qualificados permite ao usuário se preocupar apenas com o processo analítico dos dados, e desta forma, extrair informações e relatórios, como proposto.

Apesar das limitações que a API do Twitter impõe, ainda é possível criar aplicações interessantes, usando os métodos corretos para a estrutura e análise dos

[1] "Eleições 2018 levam ódio e desavença às relações - Política - Estadão." 30 set. 2018, Disponível em: https://politica.estadao.com.br/noticias/eleicoes,eleicoes-2018-levam-odio-e-desavenca-as-relacoes,70002525774. Acesso em: 18 mar. 2019.

dados. A exploração do modelo multidimensional do Data Warehouse são só alguns exemplos do que pode ser feito.

As consultas efetuadas e os dados extraídos, foram capazes de demonstrar bem o sentimento dos eleitores a respeito das eleições como um todo e dos candidatos. Muita indiferença dos eleitores em relação às eleições; grande parte das pessoas que possuíam algum sentimento em relação aos candidatos, levaram para o Twitter o sentimento geral sobre os políticos: desaprovação, seja por ações ou ideologias de cada. O fato é que, a amostra deste estudo e sua análise é coerente até certo ponto com os fatos verificados no mundo real, gerando a necessidade de melhorias na aplicação com um todo.

Referências

AGUIAR, Erickson et. al. Análise de Sentimento em Redes Sociais para a Língua Portuguesa Utilizando Algoritmos de Classificação. 2018.

ANDRADE, Carina et. al. O Twitter como Agente Facilitador de Recolha e Interpretação de Sentimentos: Exemplo na Escolha da Palavra do Ano. In: 15ª Conferência da Associação Portuguesa de Sistemas de Informação. 2015.

ARAÚJO, Mateus et. al. Métodos para análise de sentimentos no Twitter. In:Universidade Federal de Minas Gerais. 2016.

CAMILO, Cássio et al. Mineração de dados: conceitos, tarefas, métodos e ferramentas. In: Instituto de Informática. Universidade Federal de Goiás. 2009. p 12- 15.

CAVALCANTE, Paulo Emílio Costa. Um dataset para análise de sentimentos na língua portuguesa. 2017.

CORRÊA, Igor. Análise de sentimentos expressos na rede social Twitter em relação aos filmes indicados ao Oscar 2017. In: Universidade Federal de Uberlândia. 2017.

JUNIOR, V. F., WOSZEZENKI, C., ANDERLE, D. F., SPERONI, R., NAKAYAMA, M. K. (2014). A pesquisa científica e tecnológica. Espacios, 35(9).

JUNQUEIRA, Kássio TC; DA ROCHA FERNANDES, Anita Maria. Análise de Sentimento em Redes Sociais no Idioma Português com Base em Mensagens do Twitter. Anais do Computer on the Beach, p. 681-690, 2018.

KIMBALL, Ralph; ROSS, Margy. The data warehouse toolkit: the complete guide to dimensional modeling. John Wiley & Sons, 2011.

MANSMANN, Svetlana. Building a Data Warehouse for Twitter Stream Exploration.In: University of Konstanz, Germany. 2012.

MORAES, Silvia et. al. 7x1•PT: um Corpus extraído do Twitter para Análise de Sentimentos em Língua Portuguesa. 2015.

NASCIMENTO, Paula et. al. Análise de sentimento de tweets com foco em notícias. 2012.

NOGUEIRA, Rodrigo R. Newsminer: um sistema de datawarehouse baseado em texto de notícias.In: Universidade Federal de São Carlos. 2017

PIZZANI, Luciana et. al. A arte da pesquisa bibliográfica na busca do conhecimento. In: Revista Digital de Biblioteconomia e Ciência da Informação.

TAVARES, Jonatas et al. Soluções de BI 2.0 para Análise de Dados a partir do Twitter®: Eleições 2014.2017.

TREVISAN, Allan. MINERAÇÃO DE TEXTOS NO TWITTER .2015

TSE. Disponível em <http://divulga.tse.jus.br/oficial/index.html>. Acesso em 10 dez. 2018.

Avaliação de Abordagens Probabilísticas de Extração de Tópicos em Documentos Curtos

Michel Chagas da Costa[1], **Denio Duarte**[1]

[1]Universidade Federal da Fronteira Sul
Campus Chapecó
Chapecó – SC – Brazil

costa.michel10@gmail.com, duarte@uffs.edu.br

Abstract. *Short texts are very popular in social media. Comments and reviews are examples of common short texts found in the Web. Topics extraction from text is a challenging task for content analysis. Lately, probabilistic topic modelling has been used as a tool for topic extraction. To extract topics from short documents is more challenging since the word co-occurrence is more sparse. The aim of this work is, thus, evaluate some short documents topic modelling to identify which one is more suitable in the scenarios proposed. We conduct experiments on three short text collections, and results show that the approaches have similar performances.*

Resumo. *Devido ao amplo uso das redes sociais, textos pequenos se popularizaram na Web. Extrair tópicos de uma grande quantidade de textos curtos tornou-se uma tarefa crítica e desafiadora em tarefas de análise de conteúdo. Neste contexto, várias abordagens foram propostas para inferir tópicos a partir de conjuntos de coleções de textos curtos. Este trabalho tem como objetivo avaliar o uso de algumas destas abordagens probabilísticas na extração de tópicos em documentos curtos utilizando métricas para este fim. Os experimentos realizados em três coleções mostram que as abordagens estudadas tem resultados similares nos cenários propostos.*

1. Introdução

Textos curtos dominam a Web, tanto no contexto de sites tradicionais - títulos de páginas, anúncios, legendas de imagens, mensagens em fóruns e títulos de notícias - quanto mídias sociais, que tiveram um grande crescimento, como *tweets* e mensagens de status [Cheng et al. 2014]. Há um número muito grande de textos curtos, o qual está em rápido e constante crescimento. Um exemplo disso é o *Twitter* que com 250 milhões de usuários ativos gerava aproximadamente meio bilhão de *tweets* por dia [Zuo et al. 2016a]. E este grande volume de textos curtos contém informações que dificilmente são encontradas nas fontes tradicionais de busca e que trazem informações sofisticadas do mundo real.

Abordagens probabilísticas para modelagem de tópicos têm sido usadas de modo amplo para extrair automaticamente tópicos de uma grande coleção de documentos. Abordagens usuais assumem a premissa de que um documento é gerado a partir de múltiplos tópicos. A abordagem *Latent Dirichlet Allocation* (LDA) [Blei 2012] possui a forma mais simples de modelagem de tópicos e serve como base para outras abordagens, também

assume a premissa acima citada. Apesar de se mostrar como uma abordagem de sucesso para textos grandes, como notícias, artigos científicos e blogs, abordagens clássicas, como o LDA, se mostraram limitadas quanto a textos curtos. Em essência, elas descobrem tópicos capturando, implicitamente, a co-ocorrência de padrões de palavras em um documento. Como em textos curtos a co-ocorrência de padrões de palavras em um documento é algo esparso, as abordagens convencionais não são eficientes para extrair tópicos neste cenário [Cheng et al. 2014, Zuo et al. 2016a, Zuo et al. 2016b]. Dados esparsos constituem o principal problema na modelagem de tópicos em documentos curtos [Quan et al. 2015]. São necessárias, então, abordagens que se adaptaram para ter seu foco em textos curtos. Cada uma delas usa diferentes premissas para procurar resolver o problema dos dados esparsos.

Desta forma, este trabalho tem como objetivo avaliar uso de abordagens probabilísticas para a extração de tópicos em documentos curtos. Serão utilizadas quatro abordagens: *Biterm Topic Model* (BTM), *Pseudo-document Topic Model* (PTM), *Self-Aggregation based Topic Model* (SATM) e *Word Network Topic Model* (WNTM). Este trabalho avaliará os resultados da execução destas quatro abordagens sobre três conjuntos de dados através de três métricas de coerência das sete apresentadas por Röder *et al.* [Röder et al. 2015a]: C_V, C_{UMass} e C_A que representam as métricas com melhor e pior desempenhos e desempenho mediano, respectivamente. Os conjuntos de dados utilizados possuem tamanhos médios de 6, 15 e 84 palavras por documento. Cada algoritmo será executado no cenário de 30, 60 e 120 tópicos. Por fim, este trabalho apresentará os resultados avaliando o uso destas quatro abordagens probabilísticas para modelagem de tópicos em textos curtos. O objetivo da análise é identificar qual abordagem se comporta melhor em cada cenário proposto.

A próxima seção apresenta o referencial teórico. Em seguida, alguns trabalhos relacionados são apresentados. A Seção 4 apresenta o projeto e os resultados dos experimentos. Finalmente, a Seção 5 apresenta conclusão.

2. Referencial Teórico

Na área de aprendizado de máquina há uma subárea que visa extrair tópicos de uma coleção de textos. Estes tópicos podem ser usados para categorizar textos, auxiliando na definição de quais temas são abordados em um texto ou conjunto de textos. Por meio de métodos probabilísticos, esses algoritmos são a base desta subárea chamada de modelagem de tópicos [Blei 2012, Steyvers and Griffiths 2007]. A modelagem de tópico é uma técnica não supervisionada, assim métricas tradicionais como precisão, revocação e acurácia não se aplicam.

Dada uma coleção de textos não-organizados, os algoritmos de modelagem de tópicos têm como objetivo descobrir os principais conjuntos de palavras, que podem ser vistos como assuntos, relacionados à coleção. Esses algoritmos podem ser adaptados para os mais diversos tipos de dados. Entre outras aplicações, eles vêm sendo usados para descobrir padrões em dados genéticos, imagens e redes sociais [Blei 2012].

A forma mais simples de modelar tópicos é através da Alocação Latente de Dirichlet - *Latent Dirichlet Allocation* (LDA) [Blei 2012]. O LDA serve como base para vários outras abordagens de extração de tópicos, inclusive abordagens para textos curtos discutidas mais adiante.

Um texto geralmente apresenta múltiplos tópicos, tratando sobre assuntos diversos que têm ligação entre si. Um mesmo texto pode, por exemplo, tratar sobre futebol, cultura e medicina. É nesse pressuposto de que um mesmo texto pode tratar sobre uma variedade de assuntos que se apoia o LDA.

Outro pressuposto desta abordagem é que um documento é uma mistura de tópicos e um tópico é uma distribuição probabilística sobre as palavras. Por exemplo, considere as palavras "televisão" e "competição". A palavra "televisão" tem uma probabilidade pequena de aparecer em um tópico sobre esportes e tem uma probabilidade maior de aparecer em um tópico sobre eletrodomésticos. E a palavra "competição" tem uma probabilidade maior de aparecer em um tópico relacionado a esportes.

Cada documento exibe tópicos em diferentes proporções e cada palavra presente no documento está associada a um destes tópicos exibidos no documento. A alocação de tópicos por documento, de forma estatística, é feita usando a distribuição de Dirichlet [Blei 2012], o que explica o nome LDA. Cada documento, em uma coleção de documentos, compartilha os mesmos tópicos. O que muda é a proporção com que cada tópico aparece no documento.

No caso do LDA, as variáveis observadas são as palavras dos documentos e as variáveis ocultas são a estrutura de tópicos. Portanto, o problema computacional de inferir a estrutura de tópicos é o problema de computar a distribuição condicional das variáveis ocultas, dada as variáveis observadas. O cálculo da distribuição condicional é computacionalmente intratável. Geralmente, os algoritmos de modelagem de tópicos são adaptações para se aproximar da distribuição condicional (posterior).

O LDA possui algumas premissas que norteiam sua implementação. Como dito anteriormente, o LDA serve como base para outras abordagens que tem como objetivo a extração de tópicos em um conjunto de dados. Conforme o objetivo, essas outras abordagens podem relaxar algumas destas premissas, a fim de adaptar o LDA para o que seja mais interessante no contexto daquele outro modelo de tópico.

Uma das premissas é a sacola de palavras - *bag-of-words*. As palavras são vistas de modo independente, soltas [Steyvers and Griffiths 2007]. Segundo esta premissa, a ordem das palavras não importa. Isto pode ser um problema com palavras que causam ambiguidade, onde a mesma palavra tem mais de um sentido semântico (polissemia). Como exemplo, a palavra "vela", que pode ao mesmo tempo significar um barco à vela; a vela feita de cera, para iluminar; ou ainda uma conjugação do verbo velar, que significa estar vigilante. Por isso, algumas outras abordagens procuram adaptar esta premissa.

Outra premissa é que a ordem dos documentos de uma coleção também não importa. Essa premissa pode ser relaxada em outros modelos de tópicos, em que a ordem dos documentos importa, como por exemplo, ao verificar a mudança de um tópico durante uma linha de tempo. Uma abordagem que contemplaria isso é o modelo dinâmico de tópicos, que respeita a ordem dos documentos [Blei 2012].

Assume-se também que o número de tópicos é conhecido e não muda: esta é a terceira premissa do LDA. Ou seja, ao organizar uma coleção de documentos, o número de tópicos já é definido e permanece fixo. Como alternativa, o modelo Bayesiano não-parametrizado de tópicos determina o número de tópicos durante o aprendizado, quando há a inferência do posterior.

2.1. Textos curtos

Os modelos de tópicos convencionais, como LDA, conseguem modelar tópicos de forma satisfatória em uma coleção de textos longos. Porém, na Web, os textos curtos prevalecem [Cheng et al. 2014]. Títulos de páginas, anúncios, legenda de imagens, títulos de notícias, *tweets*, mensagens em redes sociais são apenas alguns exemplos da variedade de textos curtos encontrados na Web. Devido à grande quantidade de textos curtos, tornou-se importante modelar tópicos de textos curtos para várias aplicações de análise de conteúdo, como, por exemplo, descobrir o perfil de interesse do usuário.

Quanto à modelagem de tópicos em textos curtos, as abordagens como o LDA apresentam uma limitação. Nelas, o número de ocorrências de uma palavra em um documento ou uma coleção é fundamental para inferir os tópicos. Entretanto, textos curtos, devido ao seu tamanho, são muito mais esparsos em termos de ocorrência de palavras. Esse problema dos dados esparsos é o principal desafio na modelagem de tópicos em textos curtos [Quan et al. 2015].

As coleções de textos curtos demandaram algumas adaptações na modelagem de tópicos, devido aos dados esparsos. A combinação do LDA com outras técnicas resultou em novas ferramentas para modelagem de tópicos em conjuntos de dados de textos curtos. Por trás de cada cada abordagem há uma intuição básica que busca resolver o problema dos dados esparsos. Algumas destas são: agrupar pares de palavras em vez de palavras soltas (BTM); criar redes de palavras valorizando as ligações entre elas (WNTM); agregar vários textos curtos com tópicos possivelmente similares (SATM); criar textos longos a partir de textos curtos considerando que este texto longo seja híbrido (PTM). Estas abordagens são brevemente apresentadas a seguir.

BTM [Cheng et al. 2014]: o BTM extrai tópicos de textos curtos modelando a geração de termos-pares na coleção de documentos. Termo-par é um par de palavras não ordenadas em um texto curto. É uma forma de explicitar a co-ocorrência de palavras relacionadas em documentos. O BTM assume que duas palavras em um termo-par compartilham o mesmo tópico tendo em vista a coleção de documentos. Segundo Cheng et al [Cheng et al. 2014], se forem agregados todos os padrões de co-ocorrências de uma palavra no *corpus* (conjunto de exemplos), suas frequências são mais estáveis e revelam mais claramente a correlação entre as palavras.

Comparado aos modelos de tópicos convencionais, o BTM apresenta duas vantagens: (i) modelar explicitamente os padrões de co-ocorrências de uma palavra, e (ii) o BTM usa os padrões de co-ocorrência de termos-pares na coleção para descobrir tópicos, visando acabar com o problema de dados esparsos. Por exemplo, um documento com três palavras $(w_1,\ w_2,\ w_3)$ se tornaria $(w_1w_2, w_1w_3,\ w_2w_3)$

SATM [Quan et al. 2015]: o modelo de tópicos baseado em auto-agregação é motivado pela agregação de textos curtos em mídias sociais, como, por exemplo, as *hashtags*, e busca prover uma solução generalizada para extrair tópicos em textos curtos de vários tipos. A ideia da agregação é que as palavras mais usadas podem criar um cluster de textos curtos com tópicos similares, levando a uma solução para o problema dos dados esparsos.

Esta abordagem assume que cada trecho de um texto é parte de um outro texto longo que não está explícito na coleção. Durante a inferência de tópicos, há uma

integração orgânica entre a modelagem de tópicos e a auto-agregação de textos.

PTM [Zuo et al. 2016a]: movido pelo potencial dos métodos de agregação, como o SATM, para lidar com os dados esparsos, um modelo de tópicos baseado em pseudo-documento para textos curtos foi proposto por Zuo et al [Zuo et al. 2016a]. Nesta abordagem, um pseudo-documento é essencialmente um tópico híbrido que combina tópicos específicos de vários textos curtos.

A chave desta abordagem, para lidar com os dados esparsos, é a introdução de pseudo-documentos através da agregação implícita de textos curtos. Desta forma, a modelagem de tópicos de uma coleção grande e esparsa é transformada em uma coleção menor, visando melhorar a eficácia e a eficiência.

WNTM [Zuo et al. 2016b]: diferentemente de abordagens como o LDA, que modela tópicos com base na co-ocorrência de palavras dentro de um documento, o que o torna extremamente sensível ao tamanho de documentos e ao número de documentos relacionados a cada tópico, o modelo de tópico de rede de palavras baseia-se na co-ocorrência de palavras dentro de uma rede de palavras.

O WNTM foi proposto para lidar com o problema dos dados esparsos e com o desbalanceamento de documentos por tópico. A principal ideia desta abordagem vem das seguintes observações: 1) quando os textos são curtos, o espaço de palavra por documento é muito esparso, enquanto o espaço de palavra por palavras é mais denso. Então desde que a qualidade dos tópicos possa ser garantida, a escolha de uma rede de palavras em vez de uma coleção de documentos é mais razoável, 2) a distribuição de tópicos por palavras, em vez de tópicos por documento, pode revelar tópicos raros que não seriam revelados em uma abordagem que usa tópicos por documento, já que o número de palavras relacionadas a tópicos raros geralmente excede o número de documentos relacionados a estes tópicos, 3) já que a distribuição de tópicos por documentos não é aprendida de forma acurada em textos curtos ou desbalanceados, deve-se distribuir os tópicos por palavras em vez de tópicos por documentos, e 4) diferentemente de outras soluções, o WNTM visa garantir a escalabilidade em diferentes cenários.

As quatro abordagens apresentadas nesta seção terão seu uso avaliado neste trabalho, visando a extração de tópicos em conjuntos de dados de documentos curtos.

3. Trabalhos Relacionados

Os trabalhos relacionados apresentados aqui são os artigos onde as abordagens que serão utilizadas neste trabalho foram propostas. No artigo em que o BTM foi proposto por Cheng et al [Cheng et al. 2014], o LDA foi comparado com o BTM, utilizando duas coleções de documentos curtos e a métrica *PMI-Score*. Foram realizados teste com 20, 40, 60, 80 e 100 tópicos. Em todos os cenários o BTM se mostrou mais coerente do que o LDA.

O PTM e o SPTM foram comparados com outras quatro abordagens, segundo Zuo [Zuo et al. 2016a]: SATM, LDA, *Mixture of Unigrams* e *Dual Sparse Topic Model*. Para avaliação, foi utilizada a validação cruzada e 100 tópicos para todas as abordagens em todas as coleções. Em duas das quatro coleções testadas, o PTM teve melhor pontuação do que as outras abordagens. Em uma das coleções o SPTM obteve maior pontuação e SATM se mostrou melhor em um dos quatro conjunto de dados.

Quan et al [Quan et al. 2015], ao apresentar o SATM, comparam a nova abordagem proposta com o BTM e com o LDA. Foram utilizadas duas coleções e executadas estas abordagens para 50, 100, 150, 200, 250 e 300 tópicos, além de utilizar duas novas métricas apresentadas no artigo para avaliação. Os autores do artigo que apresenta o SATM, concluem que esta abordagem se mostrou masi eficiente que o BTM e o LDA naquele cenário proposto.

No trabalho da proposta da WTNM [Zuo et al. 2016b], a mesma é comparada com as abordagens BTM e LDA. Com base na validação cruzada, os autores concluem que o WNTM se mostrou melhor que o BTM e o LDA, utilizando 100 tópicos.

4. Experimentos

Foram selecionados dois conjuntos de dados: *Ohsumed*[1] e *Tag My News*[2]. O conjunto de dados *Ohsumed* consiste em títulos, autores e resumos de artigos da área de medicina, com base em 270 periódicos durante 5 anos (1987-1991). O conjunto de dados Tag My News são notícias de sites de língua inglesa obtidas através de *feeds RSS* de jornais populares.

Para cada um dos conjuntos de dados foi necessário um pré-processamento. Para o conjunto de dados *Ohsumed* foi realizado uma limpeza, de modo a deixar apenas os resumos de artigos, remover pontuações, *stopwords* e palavras de baixa frequência. Enquanto a coleção *Ohsumed* original era de 151,1 MB, após o processo de limpeza para que ficasse no arquivo apenas os resumos dos artigos, o arquivo ficou com 40,7 MB. O arquivo original contava com 155807 linhas e a média de 484 palavras por linha. O arquivo obtido após a redução e limpeza atingiu 56984 linhas e tamanho médio de 84 palavras por linha. A partir daqui, o termo *Ohsumed* será usado para se referir à coleção obtida após o pré-processamento.

O conjunto *Tag My News* também foi pré-processado. Inicialmente, o tamanho do conjunto de dados era de 11,2 MB e possuía 260832 linhas. As coleções *News-Head* (apenas as manchetes) e *News-Short* (resumo da notícia) foram geradas a partir do conjunto de dados *Tag My News* e obtiveram tamanhos de 1,4 MB e 3,7 MB, respectivamente. Ambos arquivos gerados após o processo de limpeza tiveram 32604 linhas. O conjunto de dados *News-Head* ficou, em média, com 6 palavras por linha, enquanto o *News-Short* obteve 15 palavras por linha. A Tabela 1 traz informações sobre os conjuntos de dados usados neste trabalho, inclusive propriedades que foram modificadas após o pré-processamento.

4.1. Resultados

Os experimentos foram executados em um notebook ASUSTek K45A com um processador Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz e memória RAM de 8GB. O sistema operacional utilizado foi o Linux Ubuntu 18.04.1 LTS, instalado em uma partição de 34GB. O *Java Runtime Environment*, necessário para execução dos algoritmos, rodava com a versão 10.0.2. Os algoritmos foram disponibilizados pelos autores das abordagens.

As métricas escolhidas para avaliação foram três métricas de coerência: C_V, C_{UMass}, C_A [Röder et al. 2015a], assim os resultados são avaliados em termos de coerência segundo as métricas citadas. Foi utilizada a ferramenta *Palmetto*

[1] www.mat.unical.it/OlexSuite/Datasets/SampleDataSets-about.htm
[2] http://acube.di.unipi.it/tmn-dataset

Atributo/ coleção	News-Head	News-Short	Ohsumed
Tamanho do arquivo (antes)	11,2 MB	11,2 MB	151,1 MB
Tamanho do arquivo (depois)	1,4 MB	3,7 MB	40,7 MB
Número de documentos	32604	32604	56984
Número de palavras (antes)	1340835	1340835	75405134
Número de palavras (depois)	197040	501655	4780938
Número de palavras únicas (antes)	159033	159033	155807
Número de palavras únicas (depois)	23710	37231	6982
Numero médio de palavras por documentos	6	15	84

Tabela 1. Características das coleções utilizadas neste trabalho.

(aksw.org/Projects/Palmetto.html) para calcular os resultados com base nas métricas. Essa ferrramenta utiliza uma base de dados com documentos extraídos da *Wikipedia* para avaliar os tópicos. A métrica C_V é baseada numa janela deslizante, um conjunto segmentado de *topwords*, uma confirmação indireta que usa informação mútua de pontos normalizados e similaridade do cosseno. Quanto maior o valor desta métrica, maior a coerência dos tópicos. Já a métrica C_A é baseada numa janela de contexto, uma comparação de pares de *topwords*, uma confirmação indireta que usa informação mútua de pontos normalizados e similaridade do cosseno. Quanto maior o valor desta métrica, maior a coerência dos tópicos. Por fim, a métrica C_{UMass} se baseia na contagem de co-ocorrências e uma probabilidade condicional logarítmica como medida de confirmação. Quanto maior o valor desta métrica, maior a coerência dos tópicos.

Para cada conjunto de dados foram feitas nove execuções das abordagens: três para 30 tópicos; três para 60 tópicos; e três para 120 tópicos ($i.e.$, K igual a 30, 60 e 120). A repetição de três execuções para cada cenário foi realizado a fim de mitigar a influência do fator aleatório. Os tópicos gerados foram avaliados segundo as métricas e o resultado final representa a média das três execuções considerando o tripé ”algoritmo-conjunto de dados-número de tópicos”.

Para todas as abordagens foram realizadas 100 iterações. Usou-se os hiperparâmetros indicados pelo artigo referente a cada abordagem para execução da mesma: (i) $\alpha = 50/K$ para BTM, WNTM, e $\alpha = 0.1/K$ para PTM (K corresponde ao número de tópicos), (ii) $\beta = 0.01$ para BTM, PTM e WNTM, (iii) $\alpha_2 = 0.15$ para PTM, e (iv) SATM com um *treshold* de 0.001.

A Figura 1 apresenta as top-10 palavras de um tópico escolhido a partir de $K = 120$ para todas as abordagens e coleções. Observando as palavras dos tópicos gerados, *tenis* seria o tópico para a coleção *News-Head*, *política americana* para a coleção *News-short* e *tratamento de patologia* para a coleção *Ohsumed*.

A Figura 2 apresenta o desempenho das abordagens na coleção *News-Short* para o número de tópicos definidos para a métricas C_V e C_A, respectivamente. Também apresenta a média das métricas utilizando os resultados nos três diferentes K.

A Figura 3 apresenta, como na figura anterior, o desempenho das abordagens na coleção *News-Head* para o número de tópicos definidos para as métricas C_V e C_A, respectivamente, além da média geral do desempenho.

Coleção/ abordagem	BTM	PTM	SATM	WNTM
News-Head	nadal open djokovic federer final french win wozniacki round lead	nadal djokovic federer win murray beats reach rome monte advance	murray officials final kentucky madrid nadal barcelona federer advance derby	nadal djokovic federer final indian win last murray advance wells
News-Short	president obama barack us said united states secretary would obamas	president obama federal us barack friday tuesday court deal program	ollanta vote showed race percent candidate poll june election presidential	president obama barack republican obamas governor presidential white run campaign
Ohsumed	treatment therapy treated two survival three study time years weeks	patients skin tissue one treatment two three study factors patient	cells one treatment two study may less disease group patients	treatment therapy three treated two time total study symptoms survival

Figura 1. Exemplo de tópicos gerados após a execução dos algoritmos (K=120).

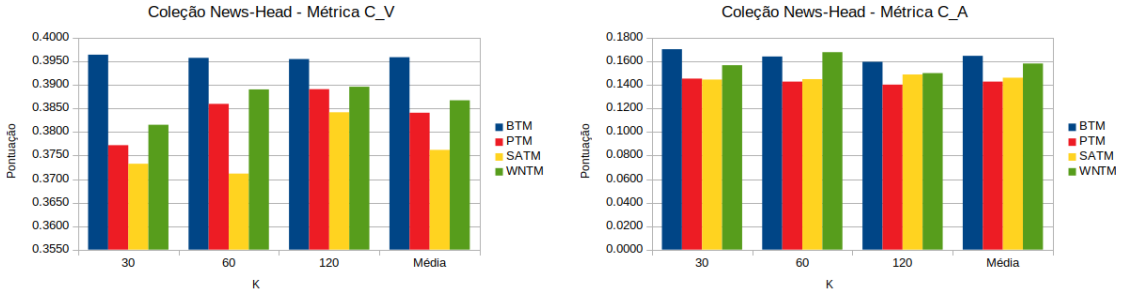


Figura 2. Resultados da C_V e C_A nos tópicos da coleção News-Head

.

Os resultados das métricas dos tópicos gerados pelas abordagens na coleção *Ohsumed* são apresentados na Figura 4. Finalmente, a Tabela 2 apresenta o desempenho das abordagens baseado na métrica C_{umass}.

Desempenho das abordagens utilizando a Métrica C_{UMass}									
Abordagem	**News-Head**			**News-Short**			**Ohsumed**		
	30	60	120	30	60	120	30	60	120
BTM	-3.34	-3.28	-3.41	-3.62	-3.78	-3.63	-3.79	-3.83	-3.98
PTM	-3.17	-3.49	-3.80	-3.46	-3.28	-3.68	-3.21	-3.34	-3.66
SATM	-3.06	-3.18	-3.66	-3.18	-3.49	-3.47	-1.56	-1.72	-2.10
WNTM	-2.87	-2.98	-3.42	-2.96	-3.17	-3.43	-3.91	-3.99	-3.97

Tabela 2. Desempenhos com a métrica C_{UMass}

As abordagens obtiveram um desempenho similar na maioria dos cenários e número de tópicos Foram distintas as abordagem que ficaram melhor ranqueadas em cada cenário. Considerando o quadro geral, o BTM e o PTM obtiveram mais coerência quando as métricas usadas foram C_V e C_A. Por outro lado, o SATM e o WNTM se mostraram mais coerentes quando a métrica usada foi a C_{UMass}. Para o quadro geral, considerando métricas, conjuntos de dados e número de tópicos, as execuções do BTM obtiveram os resultados mais coerentes em mais cenários do que as outras abordagens. A coerência

Outro ponto notado foi que, em geral, as abordagens usadas neste trabalho obtive-

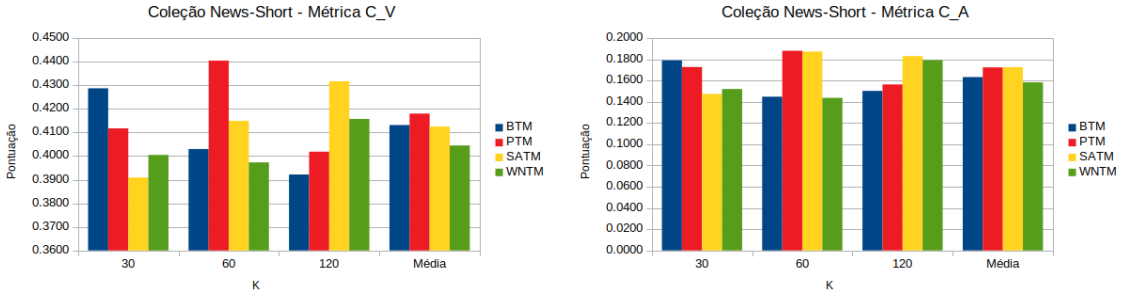


Figura 3. Resultados da C_V e C_A nos tópicos da coleção News-Short

.

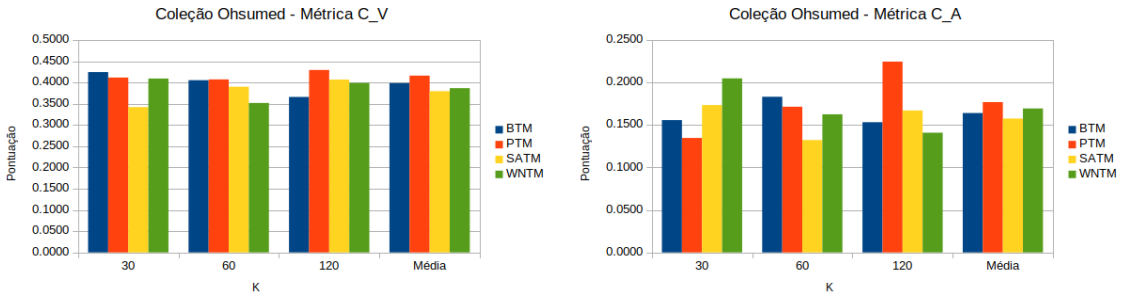


Figura 4. Resultados da C_V e C_A nos tópicos da coleção Ohsumed

.

ram melhor resposta com 30 ou 60 tópicos para as duas coleções com menor número de palavras por documento, *News-Head* e *News-Short*, com média de 6 e 15 palavras por documento, respectivamente, e obtiveram melhor resposta para o maior número de tópicos (120) com a coleção de maior número médio de palavras por documento, *Ohsumed*, com número médio de 84 palavras por documento.

5. Conclusão

Devido à dominância de textos curtos na Web, extrair tópicos de documentos curtos tornou-se uma tarefa cada vez mais importante e desafiadora. Com a ascensão das redes sociais na Web o número de textos aumentou consideravelmente. Em 2016, meio bilhão de *tweets* eram gerados por dia. Entretanto, devido a falta de co-ocorrência de palavras em coleções de documentos curtos, foi necessário o surgimento de novas abordagens, a fim de superar o problema dos dados esparsos em conjuntos de dados de textos curtos. Essas abordagens usaram diferentes premissas para atingir a finalidade de extrair tópicos de documentos curtos de modo coerente.

O BTM se apoiou na ideia de que se duas palavras que aparecem em um mesmo contexto fossem agrupadas (”termo-par”) e houvesse co-ocorrência de termos-pares na coleção, isso indicaria maior probabilidade destas duas palavras pertencerem ao mesmo tópico. O PTM baseou-se na premissa de que documentos de textos curtos pertencem a um pseudo-documento grande, mas que esse pseudo-documento grande era composto de vários tópicos distintos. Assim como o PTM, o SATM se respaldou no conceito de que pequenos textos formam um pseudo-documento grande, mas com uma distinção fundamental com relação ao PTM: para o SATM, cada pseudo-documento era composto de

pequenos documentos que integravam um único tópico. O WNTM usou como base a concepção de que era possível formar redes de palavras, ligando as palavras que aparecem próximas, como se fosse um grafo, e então gerando um pseudo-documento para assim diminuir o problema dos dados esparsos e desbalanceados.

Este trabalho avaliou o uso destas quatro abordagens que surgiram visando resolver o problema dos dados esparsos e possibilitar a extração de tópicos em conjuntos de dados de textos curtos de um modo coerente. Para isso, cenários diferentes foram apresentados, variando no número de tópicos (30, 60 e 120) e no número médio de palavras por documento (6, 15, 84).

Considerando o quadro geral, o BTM foi a abordagem que mais superou as outras na maior quantidade de casos. Apoiado pelas métricas C_V e C_A o BTM foi a que teve maior pontuação média nas execuções sobre os dois conjuntos de dados com menor número médio de palavras por documento (6 e 15) e o PTM foi a abordagem melhor ranqueada sobre a coleção com maior número médio de palavras por documento (84). A métrica C_{UMass} apontou como melhor abordagem o WNTM, se tratando dos dois conjuntos de dados com menor número médio de palavras por documento (6 e 15), e o SATM, referindo-se à coleção de documentos com maior número médio de palavras por documento (84).

Pode-se citar algumas direções para trabalhos futuros: (i) utilizar todas as métricas propostas em [Röder et al. 2015b] e avaliar a correlação entre os resultados das mesmas e (ii) utilizar uma abordagem não paramétrica para extração dos tópicos (K é calculado automaticamente).

Referências

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4).

Cheng, X., Yan, X., Lan, Y., and Guo, J. (2014). Btm: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):2928–2941.

Quan, X., Kit, C., Ge, Y., and Pan, S. J. (2015). Short and sparse text topic modeling via self-aggregation. In *IJCAI*, pages 2270–2276.

Röder, M., Both, A., and Hinneburg, A. (2015a). Exploring the space of topic coherence measures. In *Proceedings of the eight International Conference on Web Search and Data Mining, Shanghai, February 2-6*.

Röder, M., Both, A., and Hinneburg, A. (2015b). Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408. ACM.

Steyvers, M. and Griffiths, T. (2007). Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440.

Zuo, Y., Wu, J., Zhang, H., Lin, H., Wang, F., Xu, K., and Xiong, H. (2016a). Topic modeling of short texts: A pseudo-document view. In *Proceedings of the 22nd ACM SIGKDD*, pages 2105–2114. ACM.

Zuo, Y., Zhao, J., and Xu, K. (2016b). Word network topic model: a simple but general solution for short and imbalanced texts. *Knowledge and Information Systems*, 48(2):379–398.

Extração de característica para identificação de discurso de ódio em documentos

Cleiton Lima[1], **Guilherme Dal Bianco**[1]

[1]Universidade Federal da Fronteira Sul
Campus Chapecó
Chapecó – SC – Brazil

cleiton.limapin@gmail.com, guilherme.dalbianco@uffs.edu.br

Abstract. *Social media is increasingly present in people's lives, including tools that allow users to collaborate with the creation of the content. Many users utilize these functions to post texts spreading illicit or criminal content. Most works on abusive identification use supervising learning, which demands the feature extraction to achieve good quality. The meta-feature represents a state-of-the-art feature extraction on text classification. In this work, we propose a combination of feature extraction to improve the detecting of offensive speech using meta-features. Our results, on real datasets, show that our proposed combination of features outperforms in around 3.5% the effectiveness of state-of-the-art approaches.*

Resumo. *As mídias sociais estão cada vez mais presentes na vida das pessoas, incluindo ferramentas que permitam que o usuário colabore com a criação do conteúdo nelas exposto. Muitos usuários se aproveitam dessa funcionalidade para disseminar conteúdo ilícito ou criminoso. Caso não seja removido, este conteúdo será visto por cada vez mais pessoas e poderá ser propagado pela internet, atingindo um número maior de vítimas e incentivando a ocorrência de outros crimes. Este artigo propõe explorar e extrair características de textos utilizando técnicas de processamento de linguagem natural e aprendizado de máquina para detectar automaticamente discursos de ódio. Os experimentos demonstraram que o método foi capaz de melhorar a qualidade em até 3,5% em relação ao método base.*

1. Introdução

Com o advento das Redes Sociais Online (RSO), cada vez mais pessoas expõem suas ideias e opiniões nestes ambientes. Os usuários exploram aspectos de RSO, como o anonimato e políticas frágeis de publicação de conteúdo, para disseminar mensagens de discurso de ódio, como por exemplo racismo, xenofobia e homofobia, etc [Nakamura et al. 2017]. O discurso de ódio é comumente definido como qualquer comunicação que deprecie uma pessoa ou um grupo com base em alguma característica como raça, cor, etnia, gênero, orientação sexual, nacionalidade, religião ou outra característica [Nockleby 2000].

Devido à quantidade de dados que são gerados a cada dia, a auditoria manual de seu conteúdo para identificar discurso de ódio se torna uma tarefa impraticável. Filtros básicos de conteúdo, como expressões regulares ou *blacklist*, que filtram o conteúdo

de determinadas palavras, muitas vezes não fornecem uma solução adequada para a classificação [Schmidt and Wiegand 2017]. Com isso a classificação de texto - a atividade de rotular textos de linguagem natural com categorias - vem sendo aplicada em muitos contextos, desde a indexação de documentos baseada em um vocabulário controlado até a filtragem de documentos, com geração automatizada de metadados e desambiguação do sentido de palavra [Sebastiani 2002].

Através da classificação de textos é possível identificar discurso de ódio em documentos de forma automática. Para tal tarefa, métodos supervisionados de aprendizagem de máquina são aplicados para a criação de modelos que predizem se determinado documento se enquadra como discurso de ódio. Segundo [Batista et al. 2003], no aprendizado supervisionado é fornecido ao sistema de aprendizado um conjunto de exemplos $E = \{E_1, E_2, ..., E_n\}$, sendo que cada exemplo $E_i \in E$ possui um rótulo associado. O rótulo determina a qual classe o exemplo pertence. Através de um nova entrada não rotulada, o classificador é capaz de predizer a classe à qual o dado se assemelha.

A classificação de documentos utilizando aprendizado de máquina para resolver esse tipo de problema vem sendo estudada por muitas empresas que sofrem com essa adversidade, dentre as quais destacam-se o *Facebook* e *Twitter* [Nobata et al. 2016]. Para o correto funcionamento dos algoritmos de classificação de textos, é preciso que os dados possuam características informativas. Essas características ou atributos representam informações que descrevem determinado documento. Desse modo, a extração de características possibilita construir modelos de classificação que identificam se determinado documento possui ou não um discurso de ódio.

A proposta deste trabalho é extrair novas características a partir de meta-atributos gerados através de informações retiradas da vizinhança de cada documento. Inspirado no trabalho de [Canuto et al. 2016], os meta-atributos são criados através do algoritmo de classificação KNN (*k-nearest neighbors*). O KNN procura K documentos do conjunto de treinamento que estejam mais próximos deste documento com classificação desconhecida, ou seja, que tenham a menor distância. Os experimentos em duas bases de dados demostraram que a proposta obteve um ganho de até 3,5% se comparado ao método *baseline*.

O texto a seguir esta organizado da seguinte forma. Na Seção 2, são apresentados os conceitos de meta-atributos. Seção 3, os trabalhos relacionados são descritos. A proposta é apresentada na Seção 4. Os experimentos são apresentados na Seção 5. Por fim, a conclusão é descrita.

2. Meta-atributos

Meta-atributos são, em geral, manualmente projetados e extraídos de outros atributos no qual o conjunto de treinamento já é rotulado, e capturam relações fundamentais entre o par *(documento, classe)* [Canuto et al. 2016]. Os meta-atributos são capturados usando a vizinhança/similaridade de documentos previamente classificados utilizando o algoritmo de KNN para identificar os K vizinhos próximos. Os meta-atributos baseados em KNN contém os vetores de meta-atributos expressos como a concatenação dos sub-vetores descritos a seguir [Canuto et al. 2013]. Cada vetor de atributos mf é definido para um exemplo $xf \in X$ e categoria $cj \in C$ para $j = 1, 2, ..., m$. Seguem a seguir os três meta-atributos propostos no artigo:

- $\vec{v}_{\vec{x_f}}^{cnt} = [n_j]$: consiste em um vetor unidimensional (tamanho 1) dado pela contagem dos n_j vizinhos (entre os k vizinhos) de $\vec{x_f}$ que são exemplos de treino associados à determinada categoria c_j.
- $\vec{v}_{\vec{x_f}}^{ncnt} = [\frac{n_j}{n_{max}}]$: consiste em um vetor unidimensional dado pelo número n_j de vizinhos (entre os k vizinhos) de $\vec{x_f}$. O valor de n_{max} corresponde ao número de exemplos associados à classe com o maior número de exemplos dentre os vizinhos mais próximos.
- $\vec{v}_{\vec{x_f}}^{qrt} = [\cos(\vec{x}_{ej}, \vec{x_f})]$: um vetor de dimensão 5 produzido ao considerar cinco pontos que caracterizam a distribuição de distâncias de $\vec{x_f}$ para seus j vizinhos de dada categoria. As distâncias entre dois vetores $\vec{a}$ e $\vec{b}$ são computadas por similaridade do cosseno, denotada como $\cos(\vec{a}, \vec{b})$. Entre todos os pontos de distância entre $\vec{x_f}$ e seus j vizinhos de dada categoria, os cinco pontos selecionados $\cos(\vec{x}_{1j}, \vec{x_f}), \cos(\vec{x}_{2j}, \vec{x_f}), ..., \cos(\vec{x}_{5j}, \vec{x_f})$ correspondem, respectivamente, à menor distância, à maior distância, à distância média, o quartil inferior (valor que delimita os 25% dos menores pontos) e o quartil superior (valor que delimita os 25% dos maiores pontos).

Os meta-atributos descritos acima têm uma dimensão de 7 por categoria. Esse pequeno conjunto de meta-atributos é capaz de capturar informação do conjunto rotulado de três diferentes formas (conforme descrito nos itens acima). A primeira simplesmente conta o número de exemplos rotulados de cada categoria entre os k mais similares exemplos rotulados. A segunda divide o número de vizinhos em cada classe pelo número de vizinhos da classe com maior número de vizinhos, com objetivo de capturar a relação entre a classe escolhida pelo KNN (a classe com maior número de vizinhos) e as outras classes. A última informação fornecida com os meta-atributos propostos é baseada em uma análise das distâncias e distribuição das classes observada na vizinhança do exemplo. Os pontos que caracterizam essas informações são: a menor distância, a maior distância, a mediana, o quartil inferior e o quartil superior.

3. Trabalhos Relacionados

A detecção de texto abusivo vem sendo explorada com diversas abordagens. Um método bastante simplista é utilizar listas de palavras que remetem a conteúdo abusivo [Sood et al. 2012b]. Tais listas sofrem de uma baixa revocação já que o universo de termos ofensivos é bastante amplo e dinâmico. Para tentar mitigar isto, em [Sood et al. 2012a] é proposto o uso de contribuição colaborativa (ou *crowdsourcing*) para inferir termos ofensivos e dinamicamente identificar novos termos. No entanto, tais abordagens dependem de boas listas de palavras e podem resultar em uma precisão bastante reduzida.

O uso de modelos de predição (métodos supervisionados) surge como uma alternativa para possibilitar uma evolução na capacidade de identificação de textos ofensivos. Um dos primeiros trabalhos, tem como enfoque a identificação de textos ofensivos usando o método supervisionado *Support Vector Machines* (SVMs). No entanto, como os métodos supervisionados dependem do mapeamento de texto para valores numéricos, a extração de características (*features*) informativas é fundamental para alcançar uma alta qualidade. Em [Chen et al. 2012], por exemplo, usa a combinação de n-grams[1], lista de

[1]N-gram são uma sequência de termos com o comprimento de N caracteres.

termos abusivos, e manualmente constrói expressões regulares. Em [Nobata et al. 2016] são utilizados vários métodos de Processamento de Linguagem Natural (PLN) para criação de atributos. Tal trabalho propõe alguns novos atributos para aprimorar os resultados como tamanho médio de palavras, número de pontuações no documento, letras capitalizadas, entre outros.

No trabalho [PELLE; MOREIRA, 2017] é apresentado um conjunto de dados com comentários ofensivos (e não ofensivos) coletados na web brasileira. Juntamente com os dados, são apresentados resultados de algoritmos de classificação que servem como base para demais trabalhos futuros.

4. Proposta

A proposta deste trabalho é extrair novas características a partir de meta-atributos gerados através de informações retiradas da vizinhança de cada documento. Inspirado no trabalho de [Canuto et al. 2013], os meta-atributos são encontrados através do algoritmo de classificação KNN.

Para aplicação do método, inicialmente é usado o algoritmo de classificação KNN para encontrar a vizinhança mais próxima dos documentos previamente rotulados. A distância usada para determinar a proximidade dos vizinhos ao documento, é definida pela função de similaridade do cosseno.

Com as informações da vizinhança para cada documento, é feita a criação de novas características a partir das mesmas. A proposta das novas características é capturar informação do conjunto já rotulado de três diferentes formas:

1. Contagem de exemplos rotulados, da mesma maneira que faz o método KNN ao realizar a classificação;
2. Capturar a relação entre a classe escolhida pelo KNN (a classe com maior número de vizinhos) com as outras classes;
3. Análise da distribuição das distâncias para cada classe.

Na primeira são criadas duas características, que são a contagem do número de exemplos rotulados de cada categoria entre os vizinhos. Por exemplo, se 10 dos vizinhos estão classificados como discurso de ódio e os outros 20 como sendo de outra categoria, as duas novas características seriam com os valores 10 e 20.

A segunda abordagem para a criação dos meta-atributos, consiste na normalização do conjunto de meta-atributos anterior. Para tal, é feita a divisão do número de vizinhos em cada classe pela quantidade de vizinhos da classe com maior número de vizinhos. Seguindo o exemplo anterior, se 10 dos vizinhos estão classificados como discurso de ódio e os outros 20 como sendo de outra categoria, as novas características seriam os valores $10/20$ (0.5) e $20/20$ (1).

Por último, a informação fornecida com os meta-atributos propostos é baseada em uma análise das distâncias para cada classe observada na vizinhança. Para tal, foram escolhidos diferentes pontos que podem caracterizar a informação contida na distribuição das distâncias, totalizando cinco novas características por classe, sendo elas:

- **Menor distância**: Dentre todos os vizinhos, foi escolhido o vizinho mais próximo ao documento;

- **Maior distância**: De todos os vizinhos, foi escolhido o vizinho mais distante ao elemento;
- **Distância média** : De todos os vizinhos, é definida a distância média entre os mesmos.
- **Quartil inferior**: Valor que delimita os 25% das menores distâncias.
- **Quartil superior**: Valor que delimita os 25% das maiores distâncias.

5. Experimentos

Nesta seção, serão apresentados os resultados obtidos na experimentação. Serão detalhados os algoritmos que foram utilizados para a classificação dos dados, conforme descritos na proposta deste trabalho.

5.1. Configurações

A base de dados utilizada para a realização dos experimentos foi a utilizada no trabalho [de Pelle and Moreira 2017] e que pode ser obtida por *download* [2]. A base de dados é composta por duas partes denominadas *OffComBR-2* e *OffComBR-3*. As duas contêm os textos (comentários da web) juntamente com o rótulo de classificação, o qual indica se o texto representa discurso de ódio (classificação positiva) ou não. Na primeira parte, composta por 1.250 comentários, 419 destes são considerados discurso de ódio, representando aproximadamente de 33,5% e cada rótulo foi classificado por pelo menos duas pessoas. Já na segunda, são 1.033 comentários, 202 dos quais são considerados discurso de ódio, o que representa aproximadamente 19,55% dos dados e a classificação foi atribuída por três pessoas.

Para a aplicação do método, foram criados alguns conjuntos de experimentos, os mesmos propostos em [de Pelle and Moreira 2017]. Para cada experimento foram geradas determinadas características. Como é apresentado na Tabela 1, nos experimentos com o prefixo *original* foram mantidos os textos com a forma original do comentário. Já nos experimentos com prefixo *lower*, o texto foi transformado em caixa baixa, diminuindo assim a dimensionalidade das características. Alguns experimentos possuem combinações de *N-gram* (*1G*, *2G* e *3G*) e outros possuem as melhores características utilizando o ganho de informação que são apresentados com o sufixo *FS*. A coluna *LIMA* indica a quantidade de características do método proposto para cada experimento, em ambas as bases de dados *OffComBR-2* e *OffComBR-3*. As colunas *BR-2* e *BR-3* mostram o total de características referentes ao trabalho de [de Pelle and Moreira 2017] para cada base de dados *OffComBR-2* e *OffComBR-3* respectivamente. Colunas *BR-2 + LIMA* e *BR-3 + LIMA* indicam a quantidade de características da combinação dos experimentos originais com o método LIMA.

Para avaliar a eficácia do método proposto foi utilizada a mesma abordagem do trabalho [de Pelle and Moreira 2017]. A métrica avaliada para os experimentos foi *f-score*, a qual representa a média harmônica entre precisão e revocação, levando sempre em consideração o peso das classes (*f1-weighted*). Foi usada validação cruzada de dez vezes em cada conjunto de testes e feita a média do *f-score* de todas as execuções.

Os experimentos foram executados com dois algoritmos de classificação, o SVM, com os hiper parâmetros sendo: $kernel = linear$ e $C = 1.0$. E o Naïve Bayes (NB)

[2] `https://github.com/rogersdepelle/OffComBR`

Experimento	BR-2	BR-3	LIMA	BR-2 + LIMA	BR-3 + LIMA
original_1G	4.979	4.347	14	4.993	4.361
original_1G_FS	261	148	14	275	162
original_1G_2G	17.373	15.084	14	17.387	15.098
original_1G_2G_FS	263	146	14	277	160
original_1G_2G_3G	30.710	26.599	14	30.724	26.613
original_1G_2G_3G_FS	260	151	14	274	165
lower_1G	4.122	3.646	14	4.136	3.660
lower_1G_FS	259	144	14	273	158
lower_1G_2G	15.898	13.881	14	15.912	13.895
lower_1G_2G_FS	263	142	14	277	156
lower_1G_2G_3G	29.125	25.302	14	29.139	25.316
lower_1G_2G_3G_FS	268	146	14	282	160

Tabela 1. Experimentos e suas características.

que foi utilizado com os parâmetros originais do classificador. Outras combinações de configurações serão exploradas nos próximos trabalhos. Cada teste foi executado com validação cruzada de dez vezes. O número de vizinhos utilizados para extração dos meta-atributos do algoritmo KNN foi definido como 30 $N = 30$, conforme sugestão de [Canuto et al. 2016].

Para validar as comparações entre os métodos, foi utilizado o teste *Student's T-Test* (t-test) na métrica *f-score*. Esse teste é feito sobre dois conjuntos de dados e o seu resultado é um número, entre 0 e 1, que mede a confiança de uma afirmação. Neste trabalho, as afirmações que passam por validação são os resultados do trabalho [de Pelle and Moreira 2017] e da proposta deste trabalho. Caso o resultado do *t-test* for $\alpha > 0,05$, pode-se afirmar que uma proposta foi melhor ou pior que a outra em um determinado aspecto.

5.2. Execução dos Experimentos

Foram conduzidos experimentos para avaliar a eficácia e o poder discriminativo dos meta-atributos descritos anteriormente, bem como dos atributos textuais originais. Esses atributos originais serão referenciados nas tabelas e no texto como *baseline*. O grupo dos meta-atributos, método proposto neste trabalho, serão denominados como *LIMA*.

A seguir serão apresentados os resultados das execuções em ambas as base de dados. As Tabelas 2 e 3 mostram os resultados das execuções dos algoritmos de classificação SVM e NB e juntamente com o desvio padrão, a indicação de ganho estatístico de cada média representado $\uparrow$, indicação de perda estatística das médias representada por $\downarrow$ e o empate estatístico representado por $\bullet$.

Na base de dados *OffComBr-2*, após aplicar o método e suas execuções, apresentados na Tabela 2, pode-se perceber que em dois casos a média das execuções entre *baseline* e a combinação de *baseline + LIMA*, obteve-se um resultado melhor com o clas-

sificador SVM, no caso de *lower_1G_FS* teve um ganho de aproximadamente 5,14% e *lower_1G_2G_3G_FS* de 7,7%.

Na execução dos experimentos *LIMA* em relação ao *baseline*, o classificador SVM obteve resultados inferiores, no qual, somente quatro experimentos tiveram empate estatístico. Já o algoritmo de NB obteve empate estatístico em dez casos e ganho estatístico em dois, *lower_1G_FS* e *lower_1G_2G_3G_FS*.

	baseline				***LIMA***				***baseline + LIMA***			
Experimento	**SVM**	**STD**	**NB**	**STD**	**SVM**	**STD**	**NB**	**STD**	**SVM**	**STD**	**NB**	**STD**
original_1G	67,12	0,05	64,20	0,05	61,59 ↓	0,04	67,00 •	0,04	67,77 •	0,06	64,20 ↓	0,05
original_1G_FS	70,81	0,06	65,63	0,03	64,14 ↓	0,05	66,77 •	0,05	72,46 •	0,05	66,14 •	0,04
original_1G_2G	66,47	0,05	65,81	0,04	62,09 •	0,05	68,18 •	0,04	66,81 •	0,07	65,81 ↓	0,04
original_1G_2G_FS	70,05	0,06	64,15	0,03	63,08 ↓	0,03	64,37 •	0,05	71,23 •	0,05	65,83 •	0,03
original_1G_2G_3G	67,67	0,06	65,98	0,04	61,71 ↓	0,05	68,32 •	0,04	66,91 •	0,05	65,98 ↓	0,04
original_1G_2G_3G_FS	70,79	0,06	66,90	0,04	62,07 ↓	0,04	64,73 •	0,06	70,82 •	0,05	66,54 •	0,04
lower_1G	71,50	0,06	65,47	0,05	66,20 ↓	0,06	67,19 •	0,06	71,43 •	0,05	65,47 ↓	0,05
lower_1G_FS	68,66	0,06	45,80	0,08	64,57 ↓	0,05	67,57 ↑	0,05	72,19 ↑	0,05	47,02 •	0,10
lower_1G_2G	70,49	0,06	67,19	0,05	67,24 •	0,05	68,53 •	0,06	70,67 •	0,05	67,19 ↓	0,05
lower_1G_2G_FS	69,58	0,06	63,30	0,05	65,29 ↓	0,04	65,55 •	0,05	72,46 •	0,04	46,50 ↓	0,09
lower_1G_2G_3G	69,15	0,06	67,57	0,05	67,07 •	0,05	69,23 •	0,06	70,99 •	0,05	67,57 ↓	0,05
lower_1G_2G_3G_FS	66,95	0,05	41,18	0,11	67,94 •	0,04	67,22 ↑	0,04	72,11 ↑	0,05	43,20 •	0,10

Tabela 2. Experimentos com a base de dados *OffComBR-2*.

Na Tabela 3, são apresentados os resultados das execuções com a base de dados *OffComBR-3*. Destaca-se que todos os ganhos são maiores pelo fato de que a classificação dos comentários são mais precisos que a da base de dados *OffComBR-2*. Os experimentos *original_1G_2G_3G_FS*, *lower_1G_FS*, *lower_1G_2G_FS* e *lower_1G_2G_3G_FS*, tiveram um ganho estatístico com a combinação de *baseline* + *LIMA* utilizando o classificador SVM de até 5,23%.

Para o classificador NB, com a combinação *baseline* + *LIMA*, somente o experimento *lower_1G_FS* obteve melhor resultado com um ganho de 3,85%. Resultados somente com o método *LIMA*, tiveram empate estatístico em oito experimentos.

	baseline				***LIMA***				***baseline + LIMA***			
Experimento	**SVM**	**STD**	**NB**	**STD**	**SVM**	**STD**	**NB**	**STD**	**SVM**	**STD**	**NB**	**STD**
original_1G	78,16	0,03	77,82	0,07	71,73 ↓	0,00	73,73 •	0,11	78,67 •	0,04	77,82 ↓	0,07
original_1G_FS	80,61	0,03	81,07	0,02	78,78 •	0,04	77,80 ↓	0,04	81,42 •	0,04	79,97 •	0,03
original_1G_2G	78,02	0,03	77,69	0,05	71,73 ↓	0,00	76,67 •	0,03	77,86 •	0,04	77,69 ↓	0,05
original_1G_2G_FS	79,29	0,02	81,14	0,03	79,52 •	0,04	77,22 ↓	0,04	81,71 •	0,04	80,38 •	0,03
original_1G_2G_3G	77,25	0,03	77,46	0,05	71,95 ↓	0,01	76,21 •	0,03	77,44 •	0,05	77,46 ↓	0,05
original_1G_2G_3G_FS	80,19	0,02	78,67	0,03	80,49 •	0,04	69,69 ↓	0,06	82,63 ↑	0,03	79,04 •	0,03
lower_1G	77,47	0,02	76,90	0,07	71,73 ↓	0,00	77,10 •	0,04	77,11 •	0,03	76,90 ↓	0,07
lower_1G_FS	78,86	0,03	78,56	0,05	81,46 ↑	0,04	70,09 ↓	0,05	81,90 ↑	0,04	80,72 ↑	0,04
lower_1G_2G	77,62	0,04	76,69	0,04	71,73 ↓	0,00	77,26 •	0,03	78,12 •	0,04	76,69 •	0,04
lower_1G_2G_FS	79,91	0,03	78,96	0,05	78,16 •	0,04	74,17 •	0,07	82,30 ↑	0,04	77,59 ↓	0,05
lower_1G_2G_3G	77,23	0,02	76,70	0,04	72,12 ↓	0,01	76,17 •	0,04	77,91 •	0,04	76,70 ↓	0,04
lower_1G_2G_3G_FS	80,36	0,02	77,53	0,03	82,24 •	0,04	73,10 •	0,05	84,57 ↑	0,03	78,51 •	0,05

Tabela 3. Experimentos com a base de dados *OffComBR-3*.

Analisando os resultados obtidos, os meta-atributos propostos neste trabalho, ti-

veram um melhor desempenho quando somados com as características do *baseline*. O classificador com melhor desempenho foi o SVM em quase todos os ganhos estatísticos. Com o método LIMA, em alguns casos, apresentou um resultado melhor em até 3,3%. Para o classificador NB na base de dados *OffComBR-2* e *OffComBR-3*, boa parte dos resultados estivem abaixo do *baseline*.

Os experimentos que obtiveram ganhos estatísticos foram os que utilizaram redução de atributos, na qual, as características mais relevantes são selecionadas. Foi possível perceber que os meta-atributos propostos sempre estiveram presentes nos experimentos com o método de redução de atributos. Se compararmos o melhor caso do *baseline* com o melhor caso *baseline+LIMA*, podemos afirmar que o método proposto obteve ganho estatístico na base de dados OffComBR-3 e empate na base na base OffComBR-2. A Tabela 4 apresenta os resultados das duas bases de dados, somente com os experimentos que possuíram características relevantes para realizar a classificação. Levando em consideração o classificador SVM, quase todos os resultados de *baseline + LIMA* obtiveram uma média melhor que ao *baseline*.

	OffComBR-2				***OffComBR-3***			
	baseline		**baseline + LIMA**		**baseline**		**baseline + LIMA**	
Experimento	**SVM**	**NB**	**SVM**	**NB**	**SVM**	**NB**	**SVM**	**NB**
original_1G_FS	70,81%	65,63%	72,46% •	66,14% •	80,61%	81,07%	81,42% •	79,97% •
original_1G_2G_FS	70,05%	64,15%	71,23% •	65,83% •	79,29%	81,14%	81,71% •	80,38% •
original_1G_2G_3G_FS	70,79%	66,90%	70,82% •	66,54% •	80,19%	78,67%	82,63% ↑	79,04% •
lower_1G_FS	68,66%	45,80%	72,19% ↑	47,02% •	78,86%	78,56%	81,90% ↑	80,72% ↑
lower_1G_2G_FS	69,58%	63,30%	72,46% •	46,50% ↓	79,91%	78,96%	82,30% ↑	77,59% ↓
lower_1G_2G_3G_FS	66,95%	41,18%	72,11% ↑	43,20% •	80,36%	77,53%	84,57% ↑	78,51% •

Tabela 4. Experimentos com redução de atributos e seus resultados.

A partir desta análise, pode-se concluir que os meta-atributos combinados com outras características, obtiveram um bom resultado para classificação dos textos com o objetivo de identificar o discurso de ódio.

6. Conclusão

Esse artigo teve como objetivo explorar e propor novas características para a classificação de texto, com o intuito de identificar o discurso de ódio em documentos. Para tal, foram usados métodos de processamento de linguagem natural e aprendizagem de máquina.

Foi utilizado como fundamentação o método proposto por [Canuto et al. 2013], que cria meta-atributos a partir da extração de informações sobre a similaridade/vizinhança de cada documento. Tais características foram analisadas de forma isolada e em conjunto com outras características de trabalhos relacionados.

Utilizando a base de dados proposta por [de Pelle and Moreira 2017], experimentos foram realizados com diferentes combinações para analisar o uso dos meta-atributos em diferentes cenários. O método proposto obteve bons resultados em alguns casos. Os meta-atributos, combinados com características propostas por [de Pelle and Moreira 2017], obtiveram ganhos estatísticos de até 5,24% em comparação com as características originais.

Utilizando o classificador SVM, os meta-atributos, analisados separadamente, obtiveram resultados próximos ao original, mostrando que as novas características são promissoras para melhorar a qualidade da classificação.

Uma forma de complementar esse trabalho é explorar métodos de análise de sentimento, área onde houve bons resultados em trabalhos relacionados, e realizar a combinação com os meta-atributos. Novos experimentos serão realizados com intuito de avaliar configurações do classificador SVM, assim como, testes estatísticos complementares.

Referências

Batista, G. E. d. A. P. et al. (2003). *Pré-processamento de dados em aprendizado de máquina supervisionado*. PhD thesis, Universidade de São Paulo.

Canuto, S., Gonçalves, L. F., Salles, T., and Gonçalves, M. A. (2013). Um estudo sobre meta-atributos para classificaçao automática de texto.

Canuto, S., Gonçalves, M. A., and Benevenuto, F. (2016). Exploiting new sentiment-based meta-level features for effective sentiment analysis. In *Proceedings of the ninth ACM international conference on web search and data mining*, pages 53–62. ACM.

Chen, Y., Zhou, Y., Zhu, S., and Xu, H. (2012). Detecting offensive language in social media to protect adolescent online safety. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 71–80. IEEE.

de Pelle, R. P. and Moreira, V. P. (2017). Offensive comments in the brazilian web: a dataset and baseline results. In *6th Brazilian Workshop on Social Network Analysis and Mining (BraSNAM)*. to appear.

Nakamura, F. G. et al. (2017). Uma abordagem para identificar e monitorar haters em redes sociais online.

Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., and Chang, Y. (2016). Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153. International World Wide Web Conferences Steering Committee.

Nockleby, J. T. (2000). Hate speech. *Encyclopedia of the American constitution*, 3:1277–79.

Schmidt, A. and Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Sood, S. O., Antin, J., and Churchill, E. (2012a). Using crowdsourcing to improve profanity detection. In *2012 AAAI Spring Symposium Series*.

Sood, S. O., Churchill, E. F., and Antin, J. (2012b). Automatic identification of personal insults on social news sites. *Journal of the American Society for Information Science and Technology*, 63(2):270–285.

Acidentes nas rodovias brasileiras nos últimos 10 anos: uma análise com dados abertos

Matheus K. G. Kageyama[1], **Nádia P. Kozievitch**[1], **Rita C. G. Berardi**[1]

[1]Universidade Tecnológica Federal do Paraná (UTFPR)
Av. Sete de Setembro, 3165 - Rebouças, Curitiba - PR, 80230-901, Brazil

matheus.kageyama@gmail.com, nadiap@utfpr.edu.br, ritaberardi@utfpr.edu.br

Resumo. *Analisar dados públicos com o intuito de gerar informações relevantes à sociedade é um tema que vem sendo discutido nos últimos anos, principalmente devido à facilidade de acesso a muitos desses dados que estão disponibilizados de forma aberta na Web. Tendo isso em vista, o objetivo desse trabalho é explorar os números de acidentes de trânsito e suas características, como localização, mortalidade, tipo de via, entre outros, com o intuito de prover análises que possam auxiliar na compreensão e caracterização dessas ocorrências, utilizando-se do auxílio de tecnologias para manipulação de informações geoespaciais.*

1. Introdução

Segundo a Organização Mundial da Saúde (OMS), mais de 3400 pessoas morrem em acidentes de trânsito todos os dias e milhares ficam feridas ou incapacitadas[1]. Na classificação por causas de mortes, no ano de 2016, acidentes de trânsito ocupavam a 8ª posição no ranking mundial de mortes à frente de tuberculose e logo abaixo de diabetes, por exemplo[2].

No Brasil o número de acidentes nas rodovias brasileiras dos últimos anos também é alto, segundo dados da Polícia Rodoviária Federal (PRF)[3]. Apenas no ano 2017 foram registrados 89518 incidentes nas estradas, apesar de estatisticamente o número estar em queda quando comparado aos anos anteriores, ainda assim, em decorrência destes episódios 6245 pessoas faleceram.

Os dados apresentados acima são considerados de domínio público no Brasil e dessa forma podem ser adquiridos através de órgãos governamentais, como é o caso das informações que são utilizadas nesse trabalho (provenientes da Polícia Rodoviária Federal - PRF). A análise desses dados possibilita a resposta de perguntas como: ”quais são os locais com maiores números de acidentes?” ou ”quais os horários com maior número de acidentes?”. Dessa forma investimentos em sinalização, orientação e fiscalização podem ser melhores aplicados pelas autoridades competentes. Assim sendo este trabalho tem por objetivo analisar os dados de acidentes fornecidos pela PRF em rodovias, com o intuito de observar padrões no que se refere a característica das ocorrências, como localidade, horário, número de feridos, condições da pista, entre outros.

Na seção 2 serão abordados trabalhos relacionados, na seção 3 será descrita a metologia utilizada para análise dos dados, na seção 4 serão apresentados os resultados

[1]http://www.who.int/violence_injury_prevention/road_traffic/en/
[2]http://www.who.int/en/news-room/fact-sheets/detail/the-top-10-causes-of-death
[3]https://www.prf.gov.br/portal/dados-abertos/acidentes/acidentes

referentes a análise dos acidentes no Brasil. Na seção seguinte é apresentada a conclusão da análise com os dados de acidentes para o estado do Paraná e por último, na seção 6 é apresentada a conclusão do artigo.

2. Trabalhos relacionados

Devido ao grande número de acidentes com veículos, diversos estudos buscam compreender as causas desses eventos.

Um estudo realizado na Noruega [Gjerde et al. 2011], focou apenas na utilização de substâncias ilícitas, como álcool, drogas e medicamentos com substâncias psicoativas, e suas relações com acidentes de trânsito. Para este estudo foram utilizados dados de duas bases do governo: Estatísticas Norueguesas (contendo os dados de acidentes de trânsito) e o Instituto Norueguês de Saúde Pública (responsável pelos exames toxicológicos). O número de acidentes também foi alvo de investigação de jornais internacionais como a BBC que publicou a nota de [Kawaguti 2012] com o objetivo de identificar por meio de um estudo envolvendo entrevistas com familiares, orgãos públicos e a análise de dados de notícias e trabalhos relacionados, quais os principais motivos que levam ao alto indice de mortes nas estradas brasileiras. Os resultados apontaram que os números elevados de mortes são relacionados às más condições de direção, ocasionadas pelas más condições das rodovias; ao acesso limitado a serviços de emergência especializados; à legislação insuficiente e à inexperiência de motoristas. Concluiu-se ainda que a conscientização, a criação de leis mais rígidas e o apoio as famílias das vítimas é essencial para ajudar na diminuição destes indices.

O problema de acidentes de trânsito também foi analisado no trabalho de [Luís et al. 2016] onde os autores utilizaram a integração de dados abertos e técnicas de visualização para o transporte urbano da cidade de Curitiba-PR para diagnosticar os acidentes de trânsito na Cidade Industrial de Curitiba (CIC). Para realizar esta análise foi realizada a aquisição/caracterização de fontes de dados do SIATE (Secretaria da Segurança Pública e Administração) e do IPPUC (Instituto de Pesquisa e Planejamento Urbano de Curitiba) Por meio dos dados, descobriu-se que a maioria das colisões ocorre entre carros e motos e a maioria dos acidentados era do gênero masculino, sendo que a maioria dos acidentes era considerada grave, sem riscos para a vítima. Grande parte dos acidentes aconteceu à noite, sendo que o horário com maior número de acidentes foi das 18h às 19h59. Os pesquisadores identificaram ainda algumas estratégias como construir um banco de dados de lesões, incluir sinais de trânsito onde eles estão faltando, construir e melhorar a infra-estrutura disponível para pedestres e fornecer inspeções regulares para que os regulamentos sejam seguidos podem auxiliar na redução dos acidentes. Com o objetivo de melhorar a segurança no trânsito, o trabalho de [de Andrade L et al. 2014] mapeou as mortes no trânsito na rodovia brasileira BR277 para determinar os principais fatores ambientais que afetam as mortes no trânsito. Para realizar este mapeamento foi realizada uma análise espacial, onde foram especificados geograficamente os locais onde as colisões ocorreram e avaliados os padrões de distribuição por meio da visualização do mapa; uma análise Wavelet e uma análise de Kernel para decompor as séries de números de mortes em cada setor rodoviário da BR277; e uma análise do ambiente construído, para identificar as associações entre o ambiente construído e acidentes de trânsito. No período analisado foram notificados 379 acidentes, com 466 mortes no BR277 onde as fatalidades foram predominantemente masculinas. As duas faixas etárias com maior ocorrência de

fatalidades foram de 31 a 50 anos e de 20 a 30 anos. Nos finais de semana houve mais mortes do que nos dias de semana com o sábado tendo a maior incidência. Observou-se que a maioria dos acidentes fatais ocorreu entre 18 horas e meia-noite. Com relação às condições climáticas,o céu limpo esteve presente durante a maior parte dos acidentes fatais seguido de tempo nublado. As fatalidades por quilômetro de rodovia simples versus pista dupla encontraram um aumento no número de fatalidades em estradas de pista dupla.

No trabalho de [Jardim et al. 2017] foram avaliadas 1013 ocorrências de acidentes de trânsito ocasionados por animais nas rodovias federais do estado de Pernambuco, entre os anos de 2012 e 2014. Para obtenção das variáveis utilizou-se o banco de dados do sistema operacional da PRF, o Siger 2. Foram utilizadas as variáveis: quantitativo de acidentes por ano, tipo de veículo, rodovia de ocorrência, tipo de pista, traçado da via, condições meteorológicas, fase do dia, tipo do solo (perímetro da via), sexo, idade dos condutores e estado físico das vítimas. Como resultado obteve-se que a maior ocorrência de acidentes aconteceu no ano de 2012 (40,1%), ocasionados por automóveis (46,9%), na BR 232 (37,5%), em pista simples (72,8%) com trechos em linha reta (92,8%), em céu claro (67,4%), fase de plena noite (65,7%), e no perímetro rural (72,2%). Predominaram os condutores do sexo masculino (86,3%), com idade entre 18 a 40 anos (54,5%) e 63,3% das vítimas foram classificadas como ilesas.

Outro trabalho que analisou o problema de acidentes de trânsito foi o de [Martins and Garcez. 2017].Foram utilizados os dados contidos no site da PRF e os dados presentes no Portal Brasileiro de Dados Abertos relacionados ao estado de Pernambuco. Os dados foram unificados numa base de dados única. Além da estatística descritiva, ainda foi criado um mapa viário de Pernambuco contendo os pontos em quem ocorreram acidentes, baseados na latitude e longitude fornecidas pela base dados. Para a criação desse mapa, foi utilizado um Sistema de Informação Geográfica (SIG). O estudo ocorreu com base em um banco de dados que abordou um total de 57.542 ocorrências, envolvendo 127.708 pessoas, entre os anos de 2007 até o mês de agosto de 2015. As variáveis que foram consideradas foram: perfil dos acidentes, perfil das pessoas, perfil dos veículos, perfil dos condutores, perfil dos acidentes graves e informação geográfica. O estudo foi fundamental para um diagnóstico mais preciso sobre as variáveis envolvidas nos acidentes. Alguns destaques sobre os acidentes foram identificados: a maiorias dos acidentes ocorrem no município de Recife, mas é Caruaru quem possui a maior parte de mortes, a BR com maior número de ocorrências é a BR-101/PE, é a sexta-feira o dia mais comum para acidentes, sendo o mês de dezembro o mais crítico, a falta de atenção é a causa mais frequente dos acidentes, destacando que a maiorias das causas é oriunda de erro humano, a colisão traseira o tipo mais comum, entretanto a colisão frontal a mais grave, pessoas entre 30 a 44 anos são os que mais se envolvem, sendo homens a maioria quando se trata do sexo dos envolvidos, automóveis são os veículos que mais se envolvem, porém são as motos que possuem consequências mais graves.

Em contrapartida aos trabalhos descritos anteriormente, o presente artigo busca uma análise mais geral, abordando todos os tipos de acidentes existentes na base utilizada (PRF), qual contém número superior de tipos de acidentes utilizados em trabalhos anteriores, além de levar em consideração todo o território brasileiro o que também difere dos antecedentes.

3. Metodologia

Os dados utilizados nesse artigo estão disponíveis no formato CSV [Shafranovich 2005], no link "https://www.prf.gov.br/portal/dados-abertos/acidentes/acidentes" . Os dados foram inicialmente importados para uma base PostgreSQL [Momjian 2001].

Salienta-se que nem sempre os dados abertos estão disponíveis nos formatos ideais para o uso desejado ou estão totalmente corretos, tornando-se necessária a realização de conversões de formatos, importações dos dados e limpeza de informações inconsistentes. Como os dados provenientes da PRF não possuíam a localização latitudinal e longitudinal para as informações anteriores a 2017, foi necessário recorrer a API do OpenStreetMap[4], no intuito de adquirir esses dados, tornando possível a geração de pontos geométricos na estrutura da extensão PostGIS, para a plotagem dos dados em mapas. Para manipulação de dados geoespaciais, a extensão PostGIS [Obe and Hsu 2011], adiciona suporte a objetos geográficos que em conjunto com a utilização do banco de dados relacional PostgreSQL, possibilita a geração de mapas de calor e cálculos de distâncias vetoriais.

Listing 1. Exemplo de consulta utilizada na análise dos dados.

```
SELECT
    to_char(horario, 'HH24') hora,
    COUNT(*)
FROM acidentes_policia
WHERE ano BETWEEN 2007 AND 2017
GROUP BY hora ORDER BY hora;
```

Através da extração de dados da base da PRF com o auxílio de consultas SQL exemplificadas pelo código 1, tem-se por objetivo gerar gráficos e mapas de calor para ilustrar a distribuição dos acidentes dentro de suas diversas categorias.

4. Análise dos acidentes no Brasil

O número de acidentes em rodovias federais no Brasil tem decaído nos últimos anos conforme Figura 1, mas ainda assim no ano de 2017 foram registrados quase 100 mil acidentes, dos quais ocasionaram 6245 mortes, conforme Figura 2.

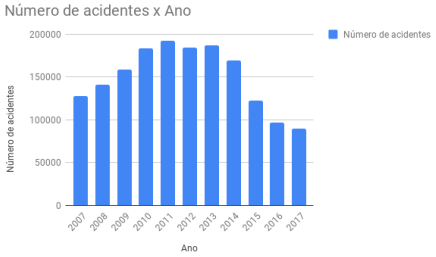


Figure 1. Número de acidentes no trânsito do Brasil por ano.

Figure 2. Número de mortes no trânsito do Brasil por ano.

Para demonstrar a importância do estudo desses dados, é possível relacionar a queda de acidentes na Figura 1 e a redução de número de mortes na Figura 2, com o

[4]https://www.openstreetmap.org

aumento de severidade na ”Lei Seca” no ano de 2012, Lei Nº 12.760/2012[5]. A mesma outorga a aplicação de penalizações aos indivíduos que forem pegos com qualquer quantidade de álcool no sangue, diferente da lei anterior que possuía uma margem de tolerância, demonstrando que punições mais severas podem ajudar na diminuição do número de acidentes e consequentemente mortes.

Com o intuito de auxiliar nas tarefas de fiscalização das estradas é possível focar nos dias com maior número de acidentes e mortes. Com isso as Figuras 4 e 3 demonstram que os finais de semana e as sexta-feiras são os dias mais perigosos para se trafegar pelas estradas, provavelmente relacionados ao fato de que são os dias mais movimentados em rodovias federais, vias comumente utilizadas para viagens.

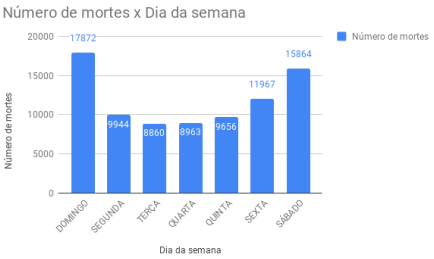


Figure 3. Número de mortes no trânsito por dia de semana.

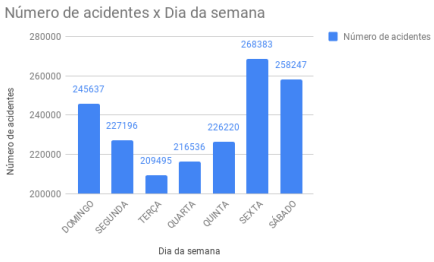


Figure 4. Número de acidentes por dia de semana.

Os horários do dia e meses também são fatores preponderantes no número de registros de ocorrências, conforme a Figura 5 e a Figura 6, os horários de ”pico”, em que ocorrem um maior número de pessoas deslocando-se entre locais como trabalho e casa é mais propício a acidentes. O mês de dezembro também registra um número relativamente maior de acidentes, provavelmente vinculado a ser um mês de férias escolares e assim muitas famílias acabam trafegando pelas rodovias em viagens.

Figure 5. Número de acidentes por horário.

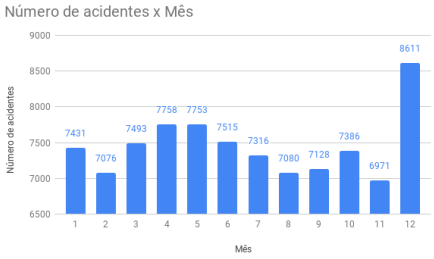


Figure 6. Número de acidentes agrupados por mês.

A orientação dos motoristas também possui papel importante na redução desse número, segundo dados ilustrados pela Figura 7, que compreende o número de acidentes classificados por suas causalidades. A falta de atenção é o maior problema enfrentado. Uma melhor educação dos motoristas quanto à importância da atenção no trânsito e direção defensiva poderia reduzir esse número.

[5]http://www.planalto.gov.br/ccivil_03/_Ato2011-2014/2012/Lei/L12760.html. Acesso em: 14 nov. 2018.

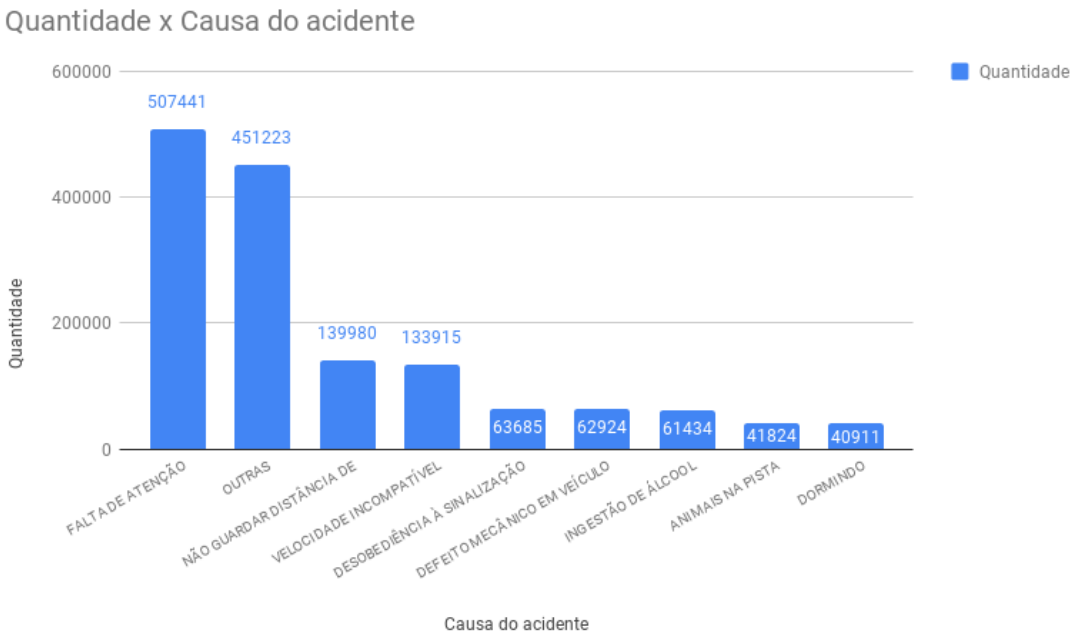


Figure 7. Maiores causas de acidentes.

4.1. Distribuição de acidentes no Brasil

Apesar dos acidentes estarem distribuídos por toda a extensão territorial brasileira, como ilustra a Figura 8, a quantidade de acidentes está concentrada em regiões específicas. Conforme a Figura 9, é possível visualizar que as regiões sudeste e sul concentram a maior parte das ocorrências registradas. Segundo dados da PRF, no ano de 2017 somente os estados da região sudeste e sul registraram 55427 registros, enquanto todos os outros registraram apenas 34091. Esses dados podem ser explicados quando se compara o tamanho da frota de veículos por região, segundo dados do Deparatamento Nacional de Trânsito (DENATRAN) de dezembro de 2017[6], as regiões sul e sudeste registravam uma frota duas vezes maior que as outras 3 juntas com 66389279 veículos contra 30702677.

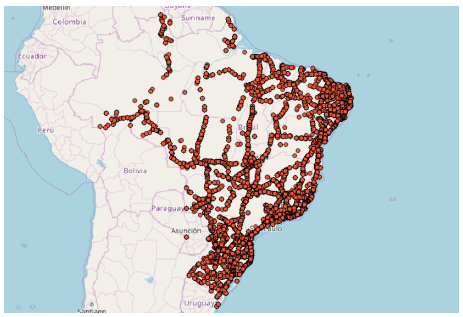


Figure 8. Pontos de acidentes no Brasil.

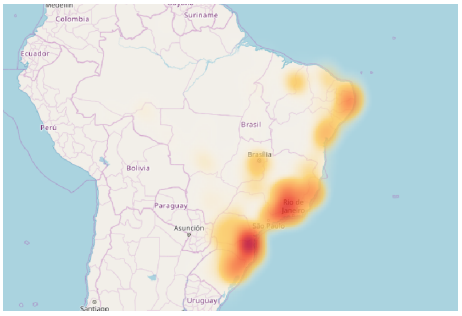


Figure 9. Mapa de calor utilizando peso por números de acidentes.

[6]http://www.denatran.gov.br/index.php/estatistica/610-frota-2017

5. Panorama dos acidentes nas rodovias federais do Paraná

Nesta seção são analisados alguns números referentes ao estado do Paraná e suas relações e comparações com os dados apresentados anteriormente na seção 4. O estado, segundo dados do DENATRAN[7], possui a terceira maior frota de veículos do País (número que ajuda a corroborar com a Figura 10) e também é o terceiro em número de acidentes.

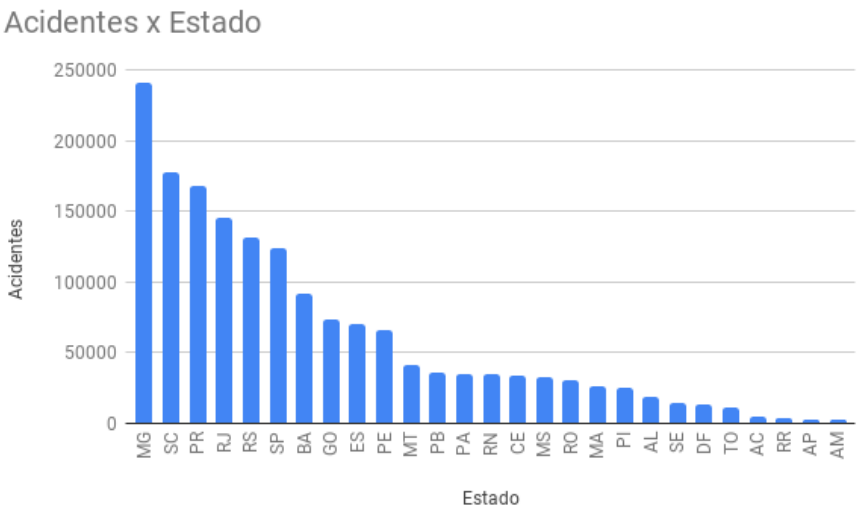


Figure 10. Acidentes por estado.

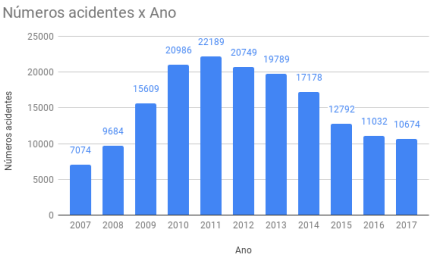


Figure 11. Acidentes no estado do Paraná por ano.

Figure 12. Acidentes no estado do Paraná por mês em 2017.

Na Figura 11, temos uma dispersão dos dados muito semelhante a apresentada na Figura 1, qual ilustra os número referentes a todo o país. Relação qual não se sustenta com a Figura 12, onde os números de acidentes por mês são muito mais relevantes no mês de Dezembro na Figura 6 do que na do estado do Paraná. Note que em relação à infraestrutura do país e do estado do Paraná, quando se analisa toda a nação percebe-se que os acidentes em pista simples acontecem mais frenquentemente que os acidentes em pista dupla ou múltipla, como indica a Figura 13. Entretanto, ao observar a Figura 14, verifica-se que os acidentes em pista dupla são muito mais frequentes no estado, sugerindo que não necessariamente uma melhor infraestrutura de pista está diretamente ligada à quantidade de acidentes.

[7]http://www.denatran.gov.br/index.php/estatistica/610-frota-2017

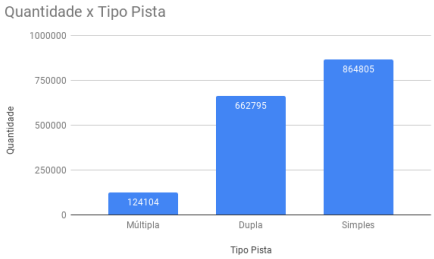


Figure 13. Acidentes por tipo de pista no Brasil.

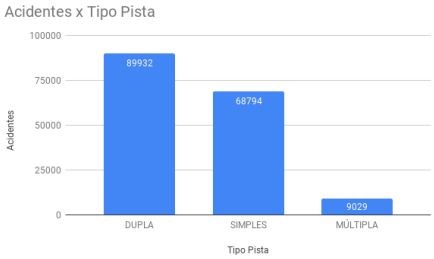


Figure 14. Acidentes no estado do Paraná por tipo de pista.

A Figura 15 apresenta a distribuição de acidentes por toda a extensão territorial, mas quando observa-se a Figura 16, percebe-se que existe uma forte concentração de acidentes em locais específicos, principalmente a capital do estado Curitiba, e as outras duas regiões mais populosas, abrangendo Londrina/Maringá e as estradas até Foz do Iguaçu, local muito frequentando por pessoas em trânsito para turismo ou acesso à fronteira com o Paraguai e Argentina.

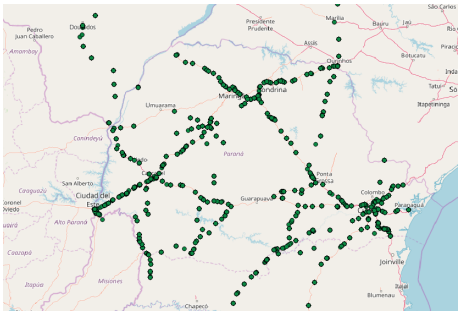


Figure 15. Localização de acidentes no Paraná.

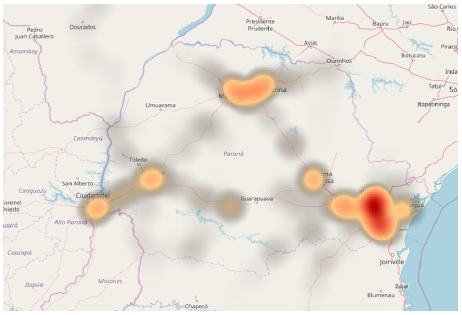


Figure 16. Mapa de calor no Paraná, com intensidade por número de acidentes.

Vale a pena notar que alguns pontos na Figura 15 não estão exatamente dentro do estado, devido ao fato de que os registros são inseridos no sistema manualmente, ocorrendo equívocos na hora do operador cadastrar as informações.

6. Conclusão

Como explorado neste trabalho, o estudo das informações disponibilizadas pelos órgãos públicos, pode gerar contribuições interessantes à sociedade. Através da manipulação de dados para visualização de informações como, meses com maior número de acidentes, localização onde os mesmos ocorrem, as principais causas e inclusive os horários mais críticos. Dessa forma ações pelos órgãos competentes podem ser julgadas, com maior conhecimento e propriedade, na tentativa de alcançar melhores benefícios e reduzir gastos ao focar nos problemas corretos.

Este trabalho apresentou uma análise genérica em termo de quantidade de categorização para os dados e também em relação ao escopo territorial, com o objetivo de servir para futuras análises comparativas entre outros estudos, através de dados que abrangem maiores categorias de acidentes e todo o território nacional.

Através das informações manipuladas neste trabalho pode-se observar que horários de pico entre 07:00 às 08:00 e 17:00 às 19:00 são horários mais propícios a ocorrência de um acidente, assim como a sua concentração histórica no mês de dezembro. Fatores como falta de atenção, imprudência de motoristas, a velocidade e a não manutenção de uma distância mínima de segurança são fatores preponderantes nas causas de acidentes.

References

de Andrade L, JR, V., and et al., R. C. (2014). Brazilian road traffic fatalities: a spatial and environmental analysis. volume 9.

Gjerde, H., Normann, P. T., Christophersen, A. S., Samuelsen, S. O., and Mørland, J. (2011). Alcohol, psychoactive drugs and fatal road traffic accidents in norway: A case–control study. *Accident Analysis & Prevention*, 43(3):1197 – 1203.

Jardim, J. M. M., da Silva Júnior, R. A., Pascoal, I. C., da Fonseca Oliveira, A. A., and Junior, J. W. P. (2017). Análise dos acidentes de trânsito ocasionados por animais nas rodovias federais do estado de pernambuco. *Revista Medicina Veterinária*, 11(1):76 – 84.

Kawaguti, L. (2012). Brazil's struggle to cut deaths on chaotic roads. *BBC*.

Luís, I. C., Kozievitch, N. P., and Gadda, T. M. C. (2016). Traffic accident diagnosis in the last decade - a case study. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1678–1683.

Martins, M. A. and Garcez., T. V. (2017). Análise descritiva dos acidentes nas rodovias federais de pernambuco (2007-2015). In *Enegep - XXXVII ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO*.

Momjian, B. (2001). *PostgreSQL: introduction and concepts*, volume 192. Addison-Wesley New York.

Obe, R. and Hsu, L. (2011). Postgis in action. *GEOInformatics*, 14(8):30.

Shafranovich, Y. (2005). Common format and mime type for comma-separated values (csv) files.

Artigos de Aplicações e Experiências

AplicExp

REx - NoSQL Redis Schema Extraction Module*

Angelo A. Frozza[1,2], Geomar A. Schreiner[1], Bruno R. L. Machado[1], Ronaldo dos S. Mello[1]

[1]Universidade Federal de Santa Catarina (UFSC)
Campus Universitário Trindade – CP 476 – 88.040-900 – Florianópolis (SC), Brasil

[2]Instituto Federal Catarinense (IFC) - Campus Camboriú
Rua Joaquim Garcia, S/N – 88.340-055 – Camboriú (SC), Brasil

angelo.frozza@ifc.edu.br, geomarschreiner@gmail.com

brunoh.rafael.leal@gmail.com, r.mello@ufsc.br

Abstract. *This paper describes the REx (**R**edis Schema **Ex**traction) module, which allows schema generation for NoSQL key-value databases, and it is coupled to the JSON Schema Discovery tool. REx implements the first step (Generation of Raw Schemas) of the schema extraction process developed in JSON Schema Discovery. The extraction strategy accomplished by REx is the main contribution of this paper.*

Resumo. *Este artigo descreve o módulo REx (**R**edis Schema **Ex**traction), que permite a geração de esquemas para bancos de dados NoSQL chave-valor, sendo um componente da ferramenta JSON Schema Discovery. REx implementa a primeira etapa (Geração de Esquemas Brutos) do processo de extração de esquemas realizado pela JSON Schema Discovery. A estratégia de extração adotada pelo REx é inovadora na área de extração de esquemas de dados NoSQL.*

1. Introdução

Na atual era do Big Data há a produção de grandes volumes de dados, em uma velocidade muito alta, armazenados de forma distribuída e compartilhados em diferentes formatos por vários tipos de aplicação [Lomotey and Deters 2014]. Entretanto, bancos de dados relacionais (BDRs) não são adequados ao gerenciamento de Big Data, principalmente por causa da rigidez dos seus esquemas de dados [Chickerur et al. 2015]. Empresas como Google e Amazon foram pioneiras no desenvolvimento de novas tecnologias para o gerenciamento de Big Data, sendo uma destas famílias de tecnologias denominadas *NoSQL (Not-Only SQL)* [NoSQL 2019]. Sistemas de BD NoSQL são capazes de representar dados complexos, são escaláveis para gerenciar grandes conjuntos de dados, adotam modelos de dados não-relacionais e, geralmente, não exigem esquemas para os dados (*schemaless*) [Sadalage and Fowler 2013].

Apesar de o esquema não ser mandatório para BDs NoSQL, a validação de dados é um requisito importante para aplicações de Big Data com alguma consistência. Assim, o gerenciamento de esquemas NoSQL torna-se um tópico pertinente para a integração

*Este trabalho foi desenvolvido com o apoio de bolsa de IC (Ed. PROPESQ 01/2017 PIBIC/CNPq UFSC) e bolsas de doutorado (PRODOUTORAL-Ed. 231/2017/REITORIA/IFC CAPES e CAPES/UFSC).

de dados, interoperabilidade entre diferentes bases de dados (até mesmo entre diferentes modelos de BDs NoSQL) ou mesmo para validação e manutenção da integridade dos dados [Scavuzzo et al. 2014, Klettke et al. 2015, Ruiz et al. 2015]. Neste contexto, foi desenvolvida a ferramenta JSON *Schema Discovery* (JSD) [Frozza et al. 2018], que realiza a extração de esquemas de coleções de documentos JSON, que é o formato de armazenamento tipicamente adotado por BDs NoSQL que seguem o modelo de dados de documento. O processo de extração gera um único esquema que representa a estrutura completa da coleção no formato JSON *Schema*. Este artigo estende as funcionalidades da JSD apresentando um módulo denominado REx - NoSQL ***R**edis Schema **Ex**traction*, que dá suporte à extração de esquemas de BDs NoSQL baseados no modelo de dados chave-valor (*Redis*). Até onde se conhece, não há trabalhos relacionados propondo a extração de esquemas de BDs NoSQL chave-valor.

O restante deste artigo está organizado conforme segue. A Seção 2 apresenta algumas informações preliminares, a Seção 3 descreve o módulo REx e como ele está integrado à JSD e, por fim, as considerações finais encontram-se na Seção 4.

2. Preliminares

O padrão JSON (*JavaScript Object Notation*) destaca-se como modelo de dados de BD NoSQL de documentos [T. Bray 2014]. Ele permite a definição de objetos cujos atributos possuem domínios atômicos ou estruturados, além de fornecer uma representação de dados flexível, pois os atributos de um objeto JSON podem ser opcionais ou de valor múltiplo. Ainda, um atributo estruturado pode conter, de maneira recursiva, um conjunto de objetos aninhados, o que é adequado à representação de entidades complexas.

Outros modelos de dados NoSQL (chave-valor, colunar e grafo) podem ser representados em JSON, uma vez que esses modelos têm um poder de expressão inferior ou equivalente ao JSON [Sadalage and Fowler 2013]. Assim sendo, pode-se afirmar que JSON serve como um formato canônico [Sheth e Larson 1990] de representação de dados de BD NoSQL com modelos de dados heterogêneos. No caso específico do modelo de dados chave-valor, cada instância pode ser representada como um atributo JSON, sendo que a *chave* do dado é o nome do atributo e o *valor* do dado é o conteúdo do atributo. Também pode-se mapear a chave do dado para o ID de um documento JSON e o valor de dados pode ser deserializado e convertido em pares atributo-valor no corpo deste documento. Esta segunda estratégia é viável quando o valor do dado é um conteúdo estruturado.

JSD é uma ferramenta para extração de esquemas de coleções de documentos JSON. O processo de extração percorre os documentos JSON de uma coleção e analisa suas propriedades para identificar o esquema bruto de cada documento. O esquema bruto é um documento JSON com a mesma estrutura hierárquica do documento JSON original, no entanto, os valores dos atributos são substituídos pelos tipos de dados correspondentes (Etapa 1 - Geração do esquema bruto). Em seguida, JSD unifica esses esquemas brutos e gera um único esquema, no formato JSON *Schema*, o qual representa a coleção de documentos como um todo (Etapa 2 - Agrupamento e unificação de esquemas brutos). Para unificar as informações estruturais de todos os esquemas brutos, uma estrutura de dados em árvore é definida, a qual mantém propriedades como: objetos aninhados, *arrays*, tipos de dados JSON primitivos ou JSON estendido. A JSD e sua abordagem de extração foi alvo de um trabalho anterior realizado por este grupo de pesquisa [Frozza et al. 2018].

A JSD foi originalmente concebida para a extração de esquemas do *MongoDB*, um BD NoSQL orientado a documentos. Recentemente, decidiu-se estender a ferramenta para lidar com a extração de esquemas de outros modelos de dados NoSQL. Para tanto, foram necessárias adaptações na sua Etapa 1. Essa adaptação resultou no módulo REx, para o caso do modelo de dados chave-valor, que é descrito na próxima seção.

3. O módulo REx

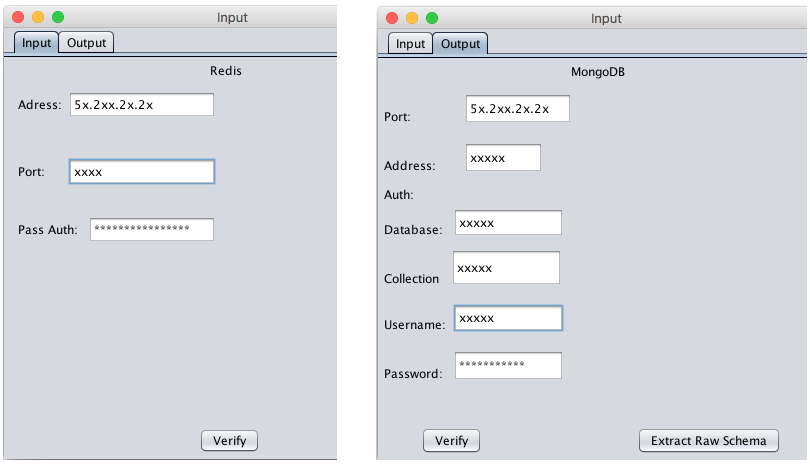
O módulo **REx** (*NoSQL **R**edis Schema **Ex**traction*)[1] é responsável pela geração do esquema bruto, no formato JSON, para instâncias de dados de um BD NoSQL chave-valor (no caso, o BD *Redis*). Para tanto, considerando que uma instância de dados em um BD chave-valor possui apenas os conceitos *key* e *value*, de acordo com o conteúdo do campo *value* a geração do esquema bruto prevê dois casos possíveis:

(a) **Documento JSON** - o conteúdo do campo *value* está estruturado no formato JSON. Neste caso, um esquema bruto para essa instância é criado considerando a estrutura hierárquica dos atributos do documento JSON. Atualmente, o REx consegue identificar seis tipos de dados JSON: *Object*, *Array*, *Number*, *Boolean*, *String* e *null*.

(b) **Sequência de *bytes*** - o conteúdo do campo *value* não é estruturado (conteúdo binário). Neste caso, o esquema bruto da instância tem apenas dois atributos: a chave *key* e o valor *value*, sendo ambos do tipo *string*.

O REx disponibiliza uma interface com o usuário, a qual é mostrada na Figura 1. A aba *Input* solicita os dados de conexão com um BD *Redis* de origem. Já a aba *Output* solicita os dados de conexão com um BD *MongoDB* no qual são armazenados os esquemas brutos produzidos. Após as informações de conexão serem verificadas, o botão *"Extract Raw Schema"* fica habilitado para iniciar o processamento.

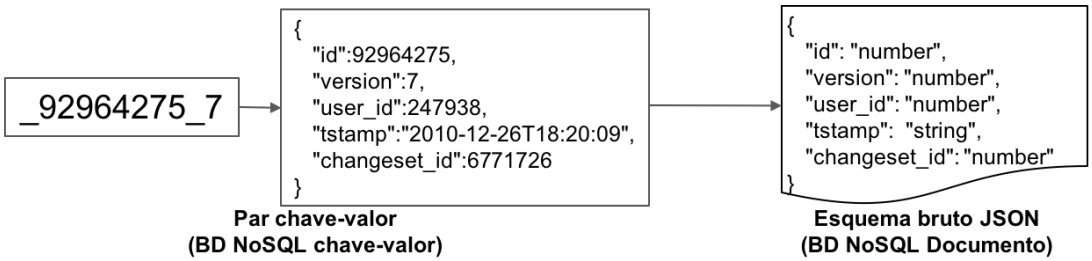
Figura 1. Interfaces do REx.



O processo de extração adotado pelo REx é o seguinte: *(i)* Carrega-se uma instância de dados do *Redis*; *(ii)* Verifica-se o conteúdo do campo *value* e procede-se a extração para cada um dos dois casos descritos anteriormente; *(iii)* Armazena-se o documento do esquema bruto gerado no *MongoDB*; *(iv)* Retorna-se ao passo *(i)*. A Figura 2 exemplifica a criação do esquema bruto a partir de um documento JSON armazenado

[1]Disponível em http://lisa.inf.ufsc.br/wiki/index.php/REx

Figura 2. Mapeamento de uma instância chave-valor para um esquema bruto.



como valor em uma instância chave-valor. Uma vez que todas as instâncias *Redis* foram processadas e foram criados os respectivos esquemas brutos, a JSD executa as suas demais etapas visando gerar o esquema único em JSON *Schema*.

4. Considerações finais

Este artigo apresenta o módulo **REx**, desenvolvido como um componente da ferramenta JSD para a extração de esquemas brutos no formato JSON *Schema* a partir de instâncias de dados presentes em BDs NoSQL chave-valor (*Redis*). Uma avaliação preliminar do funcionamento do REx concluiu que o mesmo foi capaz de atender plenamente os dois casos previstos na seção 3, desde que, no primeiro caso, a estrutura JSON esteja bem formada. Trabalhos futuros incluem o suporte à conexão com outros BDs NoSQL chave-valor, a capacidade de identificar tipos de dados geográficos e a criação de módulos para extrair esquemas dos modelos NoSQL colunar e de grafos.

Referências

Chickerur, S., Goudar, A., and Kinnerkar, A. (2015). Comparison of Relational Database with Document-Oriented Database (MongoDB) for Big Data Applications. In *8th Int. Conf. on Advanced Software Engineering and Its Applications (ASEA)*, pages 41–47.

Frozza, A. A., Mello, R. d. S., and da Costa, F. d. S. (2018). An Approach for Schema Extraction of JSON and Extended JSON Document Collections. In *IEEE International Conference on Information Reuse and Integration (IRI)*, pages 356–363. IEEE.

Klettke, M., Storl, U., and Scherzinger, S. (2015). Schema Extraction and Structural Outlier Detection for JSON-based NoSQL Data Stores. In *BTW*, volume 241 of *LNI*.

Lomotey, R. K. and Deters, R. (2014). Towards Knowledge Discovery in Big Data. In *8th IEEE International Symposium on Service Oriented System Engineering, SOSE 2014*.

Ruiz, D. S., Morales, S. F., and Molina, J. G. (2015). Inferring Versioned Schemas from NoSQL Databases and its Applications. *LNCS*, 9381:467–480.

Sadalage, P. J. and Fowler, M. (2013). *NoSQL Distilled: a Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley, 1st edition.

Scavuzzo, M., Nitto, E. D., and Ceri, S. (2014). Interoperable Data Migration Between NoSQL Columnar Databases. In *18th IEEE Int. Enterprise Distributed Object Computing Conference, EDOC, Ulm, Germany, September 1-2*, pages 154–162.

T. Bray, E. (2014). The JavaScript Object Notation (JSON) Data Interchange Format - RFC 7159.

Redes sociais intra-classe e desempenho acadêmico - uma análise inicial

Luiz Gomes-Jr[1]

[1]DAINF – UTFPR – Curitiba – PR – Brazil

gomesjr@dainf.ct.utfpr.edu.br

Resumo. *A compreensão do efeito dos diversos fatores que influenciam o desempenho acadêmico é um requisito importante para a melhoria de práticas educacionais. Como humanos são seres altamente sociáveis, fatores sociais podem ter um papel importante no contexto acadêmico. Este artigo analisa o impacto sobre o desempenho acadêmico de variáveis sociais como o círculo de amigos de um aluno e variáveis de dinâmica de grupo como tendências de agrupamentos. A análise é baseada em dados de 6 turmas da disciplina Bancos de Dados. Este artigo apresenta resultados iniciais que demonstram correlações estatisticamente relevantes entre fatores sociais e desempenho acadêmico.*

1. Introdução

A compreensão de fatores que influenciam o desempenho acadêmico é uma tarefa desafiadora. Existem múltiplas variáveis a considerar e determinar como estas variáveis afetam umas às outras e contribuem para o desempenho acadêmico é um requisito para o desenvolvimento de melhores métodos educacionais.

Um grande desafio em análise de redes sociais é a obtenção dos dados de conexões entre as pessoas. Isto é especialmente desafiador no ambiente acadêmcio, uma vez que não há um repositório estabelecido para este tipo de informação. Neste artigo empregamos relacionamentos reportados pelos alunos para a construção das redes sociais de cada turma. Pelo que sabemos, esta é a primeira vez que redes socias intra-classe são estudadas para identificar influências sobre desempenho acadêmico.

Este artigo analisa o impacto de variáveis sociais sobre o desempenho acadêmico. A análise é baseada em dados de 6 turmas da disciplina Bancos de Dados cursadas por alunos de cursos de computação. Este artigo apresenta resultados iniciais que demonstram correlações estatisticamente relevantes entre fatores sociais e desempenho acadêmico.

2. Trabalhos Relacionados

A falta de dados confiáveis para construir gráficos de redes sociais no contexto acadêmico limita a pesquisa sobre o tema. Uma abordagem para contornar o problema é inferir as relações sociais com base em outras fontes de dados. Yao et al. (Yao et al. 2017) empregaram dados de uso de cartões inteligentes por estudantes nas instalações do campus. A partir dos dados coletados, as relações sociais foram inferidas com base na co-ocorrência de eventos entre os estudantes. Os pesquisadores analisaram dados de vários locais (por exemplo, cantina, biblioteca etc.). Os dados foram usados para construir um modelo de propagação de etiquetas para prever as notas dos alunos com base nas notas de seus pares, alcançando uma precisão de cerca de 40%. Os pesquisadores não apresentam análises sobre medidas de redes sociais e seus impactos no desempenho acadêmico.

Muitas pesquisas investigam o impacto do uso de sites de redes sociais no desempenho acadêmico (ver (Doleck and Lajoie 2018) para uma revisão). O objetivo destes é determinar se o tempo gasto nesses sites pode afetar o desempenho dos alunos. Embora a maioria dos trabalhos mostre que o uso dos sites tem um impacto negativo no desempenho, a associação ainda não é um consenso.

O campo de redes complexas forneceu ferramentas e modelos para a análise de redes sociais em geral (Borgatti et al. 2009). A análise dessas redes é baseada em algoritmos que derivam medições que capturam propriedades do gráfico subjacente. Neste artigo, aplicamos várias medidas para quantificar características sociais de estudantes (nível de nó) e classes (nível de grafo). Por limitações de espaço, direcionamos os leitores para a revisão (da F. Costa et al. 2007) para mais detalhes sobre as medições utilizadas aqui.

3. Análise dos dados

Coleta e limpeza de dados: O conjunto de dados usado para a análise é baseado em dados de seis classes do tópico Bancos de Dados cursadas por estudantes de graduação entre 2016 e 2018. Os alunos são dos cursos de Engenharia de Computação e Sistemas de Informação da Universidade Tecnológica Federal do Paraná (UTFPR).

Coletar dados sociais é uma tarefa desafiadora mas, neste caso, foi simplificada pela natureza do trabalho prático especificado pelo instrutor: construir e analisar a rede social da classe. Para fornecer aos alunos os dados necessários para a tarefa, o instrutor implementou um aplicativo de rede social no qual os alunos devem inserir seus dados no início do período, incluindo suas conexões na turma.

Para cada uma das classes, o grafo de amizades foi criado e somente o maior componente conectado foi retido para representar a rede social da classe. O gráfico foi usado para computar várias medições de rede complexas. As medições foram então integradas com os dados de desempenho do aluno fornecidos pelo instrutor. Os alunos sem dados sociais ou de desempenho (provavelmente desistências) foram excluídos do conjunto de dados. O conjunto de dados final contém 148 estudantes (média de 24.7 alunos/classe, maior classe=33, menor=19).

Variáveis de turma: As variáveis de nível de turma incluídas são: agrupamento médio (avg clustering), comprimento médio do caminho mais curto (avg shortest path len), média de centralidade de intermediação (avg betweenness centr), grau médio (avg degree), assortatividade, eficiência global. As variáveis de nível de classe para as notas são: médias das provas, dos trabalhos, e da nota final.

Os resultados da análise de nível de classe têm menor relevância estatística, pois há menos dados (6 classes) para as inferências. O mapa de calor na Figura 1 mostra a correlação entre as variáveis. As correlações mostram que, em geral, quanto mais conectada é uma classe, melhores são suas notas. A correlação entre o grau médio e a nota média nas provas é de 0,75 (p = 0,087), representada na Figura 2. Existem correlações positivas entre outras variáveis que capturam a densidade do grafo, como eficiência global e agrupamento médio. A média do caminho mínimo é correlacionada negativamente com as notas por um motivo semelhante: menores caminhos mais longos aparecem em grafo menos densos. A centralidade de intermediação também é correlacionada negativamente com as notas, o que pode ser devido ao maior número de nós-ponte indicando grupos

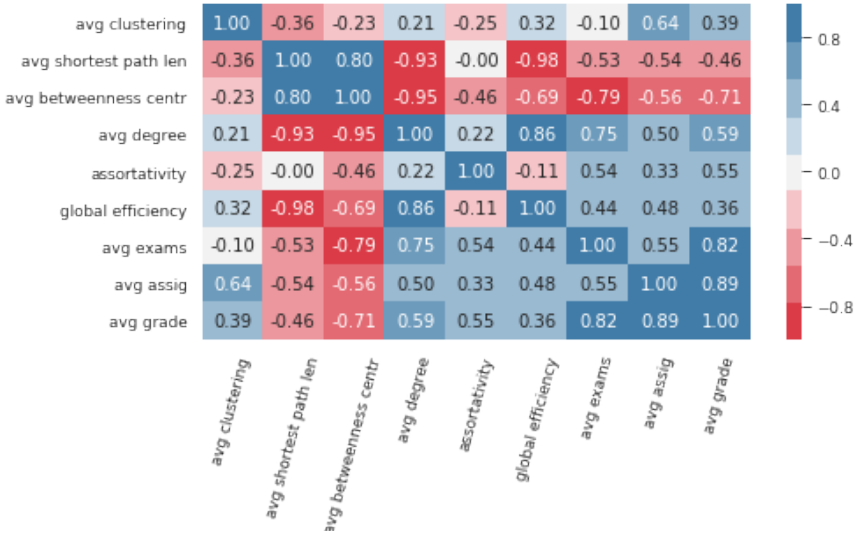


Figura 1. Correlações entre as variáveis de classe

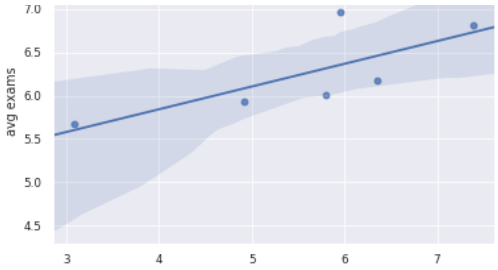


Figura 2. Média de graus (conexões) X média de nota final

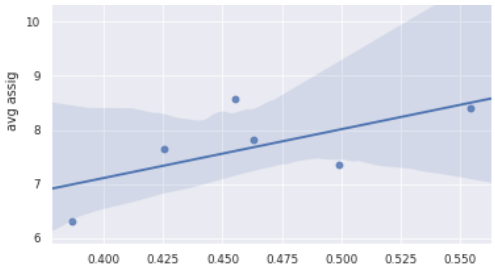


Figura 3. Média de agrupamento X média de nota de trabalhos

isolados no grafo.

Variáveis de alunos: As principais variáveis incluídas no nível de estudante são: grau médio dos vizinhos (average neighbor degree), centralidade de intermediação (betweenness), centralidade de proximidade (closeness), agrupamento (clustering), grau, centralidade de autovetor (eigenvector). Os resultados da análise de nível de aluno têm melhor relevância estatística, pois há mais dados para as inferências.

O mapa de calor na Figura 4 mostra a correlação entre as variáveis no nível de aluno. Existe uma correlação significativa ($p < 0,01$) entre a nota final dos alunos e centralidade de autovetor (correlação: 0,48) e também entre grau médio dos vizinhos (correlação: 0,43), sugerindo que alunos bem conectados tendem a ter notas mais altas. A correlação negativa (fraca) entre as notas e centralidade de intermediação pode ser devido aos estudantes que estão no meio de grupos distintos se sentirem excluídos.

A Figura 5 mostra um gráfico de alunos de acordo com a centralidade do autovetor (normalizado) e nota final. A correlação global entre as variáveis parece pertinente. Alguns padrões interessantes podem ser vistos no gráfico, como uma linha de indivíduos em torno da nota 2, provavelmente indicando alunos com alta carga de créditos. Outro padrão emerge para os alunos com baixa centralidade do autovetor, que são distribuídos regularmente entre as notas. Estes são provavelmente estudantes de outros anos ou cursos que não conheciam muitos dos seus pares. É razoável supor que os fatores sociais não

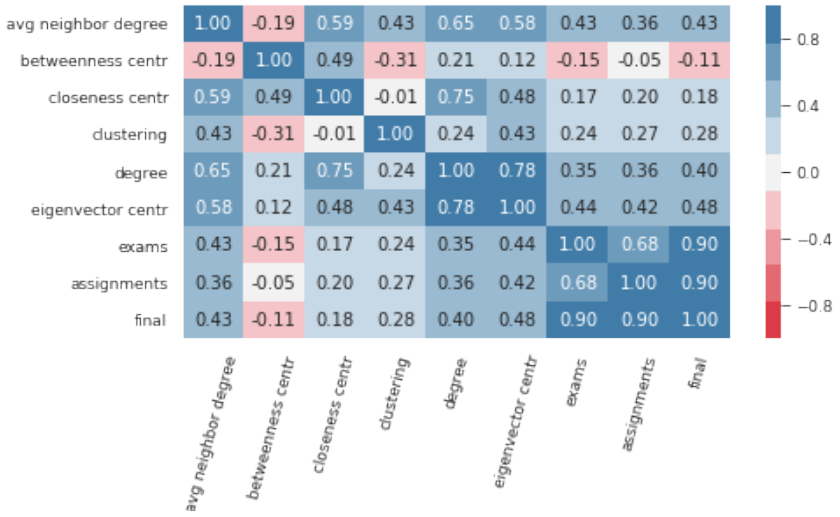


Figura 4. Correlações entre as variáveis de alunos

devem desempenhar um papel importante nesses casos anômalos.

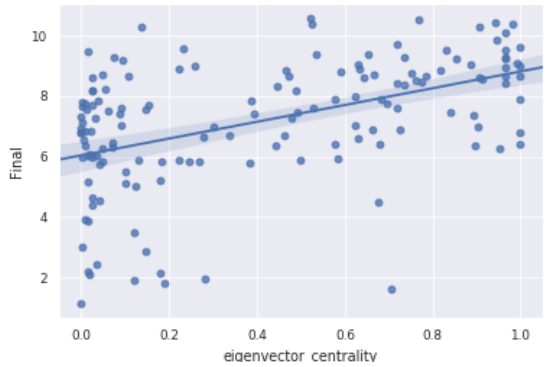


Figura 5. Centralidade de autovetor X nota final

Conclusão: Este artigo demostra resultados iniciais promissores na análise da associação entre variáveis sociais e desempenho acadêmico. Trabalhos em andamento aprofundarão a análise com a obtenção de mais dados, construção de modelos estatísticos multivariáveis, e uso de técnicas de inferência mais poderosas.

Referências

[Borgatti et al. 2009] Borgatti, S. P., Mehra, A., Brass, D. J., and Labianca, G. (2009). Network analysis in the social sciences. Science, 323(5916):892–895.

[da F. Costa et al. 2007] da F. Costa, L., Rodrigues, F. A., Travieso, G., and Boas, P. R. V. (2007). Characterization of complex networks: A survey of measurements. Advances in Physics, 56(1):167–242.

[Doleck and Lajoie 2018] Doleck, T. and Lajoie, S. P. (2018). Social networking and academic performance: A review. EAIT, 23(1):435–465.

[Yao et al. 2017] Yao, H., Nie, M., Su, H., Xia, H., and Lian, D. (2017). Predicting academic performance via semi-supervised learning with constructed campus social network. In DASFAA 2017.

Consolidação de Bases para o Diagnóstico do Distrito de Inovação de Blumenau

João Luiz Fernandes[1], Aurélio Faustino Hoppe[1], Christian Krambeck[1], Rosemeri Laurindo[1], Julio Cesar Refosco[1], Ralf Marcos Ehmke[1]

Universidade Regional de Blumenau (FURB)
Rua Antônio da Veiga, 140 - 89030-903 – Blumenau – SC – Brasil

joaooluizzf@gmail.com, {aureliof, ckrambeck, rlaurindo, refosco, ehmke}@furb.br

Abstract. *This paper presents the process of consolidating the databases of the Brazilian Institute of Geography and Statistics (IBGE), OpenStreetMaps (OSM) and the Municipality of Blumenau (PMB), with the purpose of making a diagnosis about the socioeconomic and developmental aspects of neighborhoods Victor Konder and Itoupava Seca of the city of Blumenau, being able to be adapted for the diagnosis of other regions. Such information is relevant to the understanding of the district and also considerable for the planning of policies and actions that may occur in the region considering five axes: city, people, environment, economy and mobility.*

Resumo. *Este artigo apresenta o processo de consolidação das bases de dados do Instituto Brasileiro de Geografia e Estatística (IBGE), OpenStreetMaps (OSM) e da Prefeitura Municipal de Blumenau (PMB), tendo como intuito fazer o diagnóstico quanto aos aspectos socioeconômicos e de desenvolvimento dos bairros Victor Konder e Itoupava Seca da cidade de Blumenau, podendo ser adaptado para o diagnóstico de outras regiões. Tais informações se mostram pertinentes para a compreensão do distrito e também consideráveis para o planejamento de políticas e ações que venham a ocorrer na região considerando cinco eixos: cidade, pessoas, ambiente, economia e mobilidade.*

1. Introdução

Por muitos anos, a inovação estava fortemente relacionada aos produtos tangíveis das indústrias manufatureiras, os serviços adotavam essas inovações tecnológicas, mas produziam poucas inovações em seu próprio contexto [Kon 2016]. Esta perspectiva tradicional é questionada por estudos recentes, que identificaram a intensa inovação em atividades de serviços, inclusive em setores que não apresentam fins lucrativos, como nos setores de serviços sociais e públicos. Evoluímos de um modelo de desenvolvimento baseado na produção primária e na indústria, para uma nova economia, fundamentada na informação e no conhecimento, surgiram novos arranjos e ambientes de desenvolvimento, que substituíram os antigos distritos industriais e passaram a protagonizar o processo desenvolvimento econômico e social e de geração de emprego e renda [Audy e Piqué 2016].

Estes ambientes de inovação, ou ecossistemas de inovação como são conhecidos no Brasil, são uma realidade em vários países. Como exemplo têm-se os distritos 22@Barcelona e LxFactory em Lisboa. Em território nacional existem os distritos de Pedra Branca em Palhoça, join.vale em Joinville e o Distrito C em Porto Alegre.

Para [Komninos 2008], o que distingue os ecossistemas de inovação de outras regiões é sua capacidade de reforçar o desempenho de inovação das organizações que se estabeleceram no local. Em Santa Catarina, uma das estratégias vigentes é a implantação de treze centros de Inovação, inseridos de forma descentralizada em diferentes regiões do Estado. Sendo um deles em Blumenau, servindo como pilar para ativar o ecossistema de inovação, ser referência em apoio ao empreendedorismo inovador e ser o motor da cultura inovadora [Teixeira et al. 2016]. Além disso, em 2017 o Governo do Estado publicou o Guia de Implantação dos Centros de Inovação, que apresenta os conceitos, fundamentos e diretrizes para a instalação dos Centros nas regiões catarinenses. No entanto, o guia pode servir para a implementação de qualquer habitat de inovação, já que oferece portfólios de soluções que podem ser customizados conforme a realidade de cada local [Governo do Estado de Santa Catarina 2017].

Diante do exposto, este trabalho apresenta o processo de consolidação das bases de dados do Instituto Brasileiro de Geografia e Estatística (IBGE), OpenStreetMaps (OSM) e da Prefeitura Municipal de Blumenau (PMB). Visando estabelecer um diagnóstico socioeconômico e de desenvolvimento dos bairros Victor Konder e Itoupava Seca da cidade de Blumenau, local onde será implantado o Centro de Inovação dentro do município. Tendo como intuito viabilizar a criação de um distrito de inovação na região. Os indicadores gerados também poderão ser utilizados como base para criação do Plano Estratégico de Desenvolvimento Econômico Municipal de Blumenau (PEDEN), e para avaliar o impacto das políticas sociais que serão tomadas ou negligenciadas nos próximos anos. O diagnóstico é apresentado através de um Sistema de Informação Geográfica (SIG), cujo objetivo é apoiar a manipulação, análise e visualização dos dados geográficos assim como foi feito por [Azevedo 2005].

2. Coleta, estruturação e análise dos dados

Os dados do IBGE foram obtidos através do seu portal para download. Foram coletados os dados dos censos de 2010, agregados por setores censitários e suas respectivas malhas digitais. Os dados são disponibilizados por unidade da federação, tanto em formato CSV quanto XLS, junto aos dados é necessário realizar o download da documentação contendo a explicação para cada variável das tabelas. Quanto aos dados do OSM, utilizou-se um plugin da ferramenta ArcGis. No OSM os Elementos são os componentes básicos para a reprodução do mundo, eles consistem de Nós, Caminhos e Relações, ambos podem ser associados a uma ou mais Tags, as quais descrevem significados para os elementos. Por fim, os dados da PMB, foram obtidos através de contato da FURB com a Prefeitura, foram obtidos os dados referentes a todo o município do ano de 2003 e fotos aéreas das regiões do Victor Konder e Itoupava Seca. Também foram obtidos dados do Plano Mobilidade da cidade, contendo dados sobre o transporte público, ciclovias e ciclo faixas existentes e que serão implementadas e o cadastro de lotes de 2018 para a região.

Após a coleta e análise dos dados, foi realizado a consolidação das bases utilizando o software ArcGis, pois possui todos os recursos necessários para manipulação, análise e visualização das informações. Outro ponto importante é que os dados da PMB já estavam no formato da ferramenta, além disso ela possui o plugin para integração com os dados do OSM. Após a importação dos dados, uniformizou-se o sistema de coordenadas geográficas utilizados pelas diferentes bases. Inicialmente cada uma das bases possuía um sistema de coordenadas, foi aplicado o sistema vigente para os dados da prefeitura em todas as bases, sendo ele o SAD 1969 UTM Zone 22S. Também se realizou a atualização de Tags do OSM que seriam utilizadas na geração de mapas, vinculando os edifícios do OSM e da PMB. Na sequência, foi gerado um File Geodatabase com os arquivos consolidados das diferentes bases.

A partir desse processo, foi possível realizar a geração dos indicadores utilizados para o diagnóstico do distrito. Foram utilizados os eixos definidos no ranking European Smart Cities: economia, pessoas, governança, mobilidade, ambiente e cidade. Os mapas foram gerados em formato JPEG e PDF, para os mapas gerados, foi mantido um arquivo de projeto, com a extensão .mxd, que permite a alteração e a geração de um novo mapa. A figura 1 traz os mapas de potencial construtivo (A) e cotas de enchente (B) gerados respectivamente para os eixos cidade e ambiente.

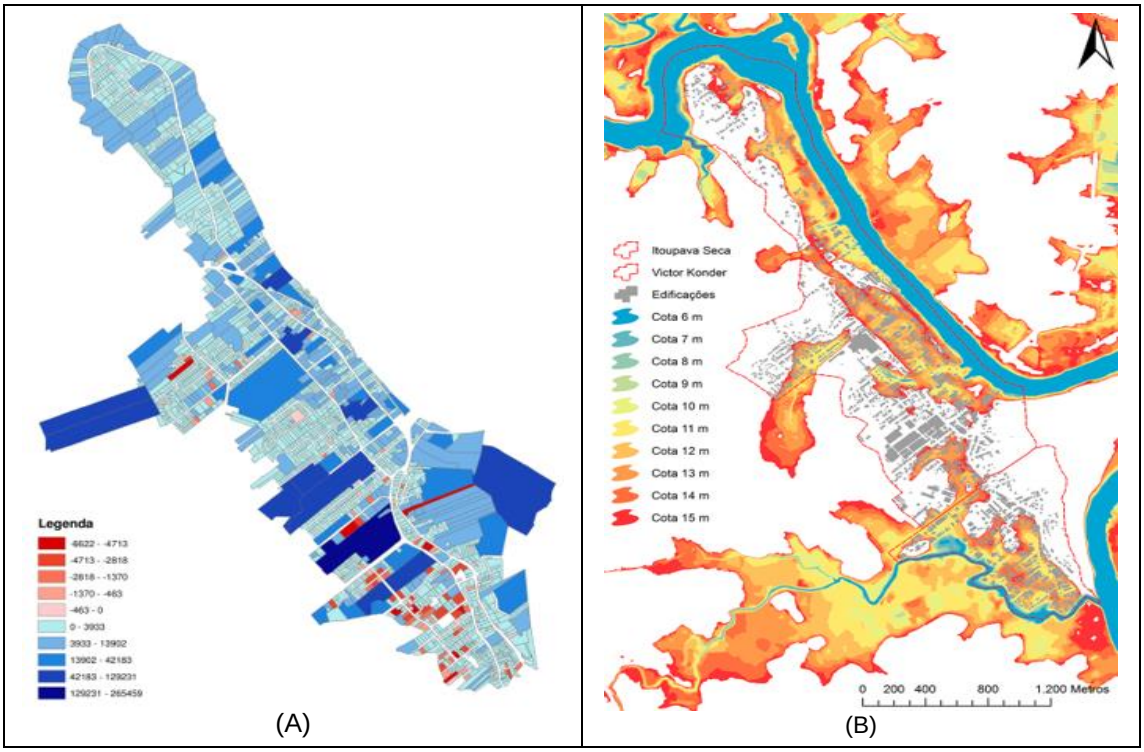


Figura 1. Mapas de Potencial construtivo (A) e Cotas de Enchente (B)

A partir do mapa de potencial construtivo (eixo cidade), foi calculado que o distrito possui hoje um total de 1 milhão de metros quadrados construídos. E possui ainda uma área de 5 milhões de metros quadrados que podem ser construídos, de acordo com o plano diretor vigente, representando um potencial de 5 vezes a área atual. No mapa, é possível observar que as áreas em azul representam o potencial construtivo, sendo as áreas mais intensa as de maior potencial. Já os trechos em vermelho, ultrapassam o limite estabelecido no plano diretor, mas não necessariamente estão fora

da legislação. Através do mapa que aponta as cotas de enchentes (eixo ambiente), foi levantado que a partir da cota de 10 metros, 18 ruas são atingidas, com a cota de 12 metros, 72 ruas são atingidas e com a cota de 14 metros, 106 são alagadas, um total de 77% da área, o que representa um dos desafios para construção do distrito. No mapa é possível observar as áreas livres de enchentes, mais indicadas para construção. E também as áreas onde será necessário lidar com os alagamentos. É importante destacar que a base desenvolvida também permita a geração de diferentes mapas não explorados nesse artigo, tais como renda per capita, população residente e densidade populacional, ocupação do solo, hierarquia viária, itinerário das linhas de ônibus, ciclovias, equipamentos urbanos (escolas, postos de saúde, assistência social, igrejas, museus), indústrias, comércios, estacionamentos, classificação das calçadas, entre outros.

3. Considerações finais

É possível concluir que a consolidação das bases de dados do Instituto Brasileiro de Geografia e Estatística (IBGE), OpenStreetMaps (OSM) e da Prefeitura Municipal de Blumenau (PMB), se mostrou uma fonte notável para o diagnóstico de aspectos socioeconômicos e de desenvolvimento. Através da base consolidada é possível diagnosticar aspectos de diferentes eixos, como cidade, pessoas, ambiente, economia, mobilidade, entre outros. As informações obtidas se mostram pertinentes para a compreensão do distrito e também consideráveis para o planejamento de políticas e ações que venham a ocorrer na região. A metodologia utilizada para a criação da base se mostra válida não apenas para os bairros do distrito, mas pode ser aplicada a qualquer outra região a fim de possibilitar o diagnóstico. A escolha da ferramenta ArcGis para a manipulação dos dados se destacou pela facilidade na integração das bases e visualização das informações, apesar disso, se mostrou necessário um analista com conhecimento na ferramenta para operacionalizar a criação dos mapas.

Referências

Audy, J., Piqué, J. Dos Parques Científicos e Tecnológicos aos Ecossistemas de Inovação. Brasilia: Anprotec, 2016. 26 p.

Azevedo, J. et al. Proposta metodológica para análise de dados socioeconômicos e ambientais para planejamento e definição de políticas públicas. Cadernos Ebape.br, [s.l.], v. 3, n. 4, p. 1-12, dez. 2005. FapUNIFESP (SciELO).

Governo do Estado de Santa Catarina. Secretaria de Estado do Desenvolvimento Econômico Sustentável. Guia de Implantação dos Centros de Inovação: Livro I-conceito e fundamentos. Florianópolis: Governo de Santa Catarina, 2017. 74 p.

Komninos, N. Intelligent Cities and Globalisation of Innovation Networks. New York: Routledge, 2008. 307 p.

Kon, A. Ecossistemas de inovação: a natureza da inovação em serviços. Revista de Administração, Contabilidade e Economia da Fundace, [s.l.], v. 7, n. 1, p. 14-27, 11 mar. 2016.

Teixeira, C., et al. Ecossistema de inovação na educação de Santa Catarina. Vieira, M. S.; Teixeira, C. S. T.; Ehlers, A. C.T.(Orgs). Educação fora da caixa, v. 2, p. 11-30, 2016.

Detecção e Análise de Metáforas usadas em *Fake News* – resultados iniciais

Guilherme Pontes Pinto[1], Leonardo de Assis da Silva[1], Luiz Filipe Kluppel Cunha [1], Luiz Gomes-Jr[1]

[1]DAINF – UTFPR – Curitiba – PR – Brasil

{guilherme.2015,leosil,luizcunha}@alunos.utfpr.edu.br, lcjunior@utfpr.edu.br

Resumo. *Este artigo apresenta resultados preliminares em detecção automática de metáforas e análise do seu uso em textos do domínio jornalístico. Para a anotação das metáforas nas notícias, empregamos um modelo de rede neural recorrente bidirecional LSTM. O objetivo principal é analisar a existência de diferenças nas características de notícias falsas em relação à notícias confiáveis no uso de linguagem metafórica. Neste artigo apresentamos resultados iniciais da anotação e análise com foco nos títulos das notícias de um corpus de artigos em inglês.*

1. Introdução

Notícias falsas, ou *fake news*, ganharam notoriedade nos últimos anos devido à influência em processos eleitorais de diversos países. Além da área política, notícias falsas são ainda frequentemente associadas a entendimentos equivocados em relação a doenças, vacinas, etc. Dada a gravidade dos problemas gerados a partir da disseminação de desinformação, amplia-se o interesse em técnicas para a detecção automática de notícias falsas de forma a analisar o enorme volume de notícias disponíveis online [Lazer et al. 2018].

Uma alternativa ainda pouca explorada diz respeito a análise das figuras de linguagem empregadas nas notícias, particularmente metáforas. Metáforas podem ser definidas como uma figura de linguagem que busca associar ideias através da comparação das mesmas, de forma a criar, na frase, um sentido não literal. De forma geral, uma metáfora pode ser definida como uma ideia explicada em termos de outro conceito, que possui certas características equivalentes com a primeira.

O uso de metáforas permeia a linguagem humana; sua aplicação pode variar do contexto de uma conversa casual ao raciocínio empregado na resolução de problemas complexos. Além disso, conforme estudos nos campos de psicologia e ciências cognitivas, o uso de metáforas pode influenciar a maneira como pessoas criam estruturas conceituais para a resolução de problemas abstratos e concretos – [H. Thibodeau and Boroditsky 2011] – isto é, características de experiências anteriores não apenas podem ser reaproveitadas em novas situações, como podem inserir um viés na maneira como novas informações são interpretadas.

Seguindo tais resultados, supomos que o padrão de uso de metáforas em notícias poderia fornecer um indicativo quanto a intencionalidade do autor em influenciar a maneira como os leitores recebem a informação. Conforme as hipóteses estabelecidas em [Horne and Adali 2017], autores tendem a utilizar dois tipos de persuasão de acordo com o tipo da notícia. Enquanto notícias confiáveis tendem convencer os leitores através de

argumentos lógicos mais extensos usando linguagem técnica, notícias falsas costumam recorrer a associações em textos curtos e repetitivos. A constatação de que notícias falsas empregam um maior número de metáforas, por exemplo, poderia indicar uma tentativa de associar conceitos negativos ao tópico alvo. Desta forma, notícias poderiam ser classificadas baseadas em seu padrão de uso de metáforas. Este artigo apresenta resultados iniciais da detecção de metáforas em notícias, focando inicialmente na análise dos títulos.

2. Trabalhos Correlatos

Embora metáforas, ao melhor de nosso conhecimento, ainda não tenham sido exploradas como aplicação no domínio de notícias falsas, diversos estudos têm pesquisado técnicas computacionais para a detecção automática de metáforas, seja através de conhecimento linguístico especializado ou modelos de redes neurais.

Particularmente, na competição descrita em [Wee Leong et al. 2018] foi comparado o desempenho de vários modelos computacionais capazes de detectar uso metafórico de linguagem à nível de palavra em textos extraídos do British National Corpus (BNC). Como *baseline* foram usados modelos baseados em *WordNet*, níveis de concretude/abstração de palavras e outras *features* adicionais. Todos os modelos que superaram os *baselines* exploraram *word embeddings*. Os sistemas de detecção de metáforas apresentaram diferença de desempenho dependendo da subcategoria textual analisada, o que indica a existência de uma disparidade no tipo e uso de linguagem metafórica de acordo com a categoria do texto, por exemplo, a frequência e tipos de metáforas exibidas em textos acadêmicos não necessariamente correspondem a padrões encontrados em notícias.

Um dos modelos que obtiveram melhor resultado, [Stemle and Onysko 2018], aplicou uma rede neural recorrente bidirecional, tipo *Long Short-Term Memory* (LSTM), seguindo a suposição de que a proficiência no uso de linguagem metafórica varia dependendo do grau de conhecimento do idioma. Dessa forma, o treinamento da rede neural responsável por aprender as *word embeddings* foi realizado através de corpus de diferentes níveis de proficiência na língua inglesa.

Em relação a modelos de regras que exploram conhecimentos especializados sobre propriedades linguísticas, como a violação de restrições selecionais, tais abordagens são limitadas a casos específicos, alcançando bons resultados somente em determinados domínios, ou metáforas compostas apenas por um pequeno subconjunto de classes gramaticais, ou idiomas que possuem características gramaticais similares. Como nosso trabalho explora uma arquitetura conexionista, isto é, seu desempenho está diretamente ligado à variáveis como topologia, parâmetros e dados de treinamento em vez de regras específicas, este poderia ser adaptado, por exemplo, para outros idiomas e contextos conforme os textos utilizados no treinamento. Ao melhor de nosso conhecimento, o presente estudo trata-se do primeiro a buscar aplicar um detector de metáforas para análise de uso e impactos em um domínio específico, no caso detecção de *fake news*.

3. Modelo

Redes Neurais Recorrentes (RNN) podem ser definidas como uma extensão das redes neurais convencionais *feedforward* tornadas em grafos cíclicos ao introduzir aresta recorrentes entre cada passo de tempo. Segundo [Lipton 2015], diferente das arquiteturas

feedforward onde o estado da rede é perdido a cada iteração, nas RNNs ocorre a transferência direta de influência de uma entrada para a(s) seguinte(s) de acordo com a força de sua ativação. A arquitetura LSTM trata-se de uma adaptação de RNN convencional na qual a camada escondida não representa mais um nó simples, mas sim uma ou mais células de memória, que mantêm um valor a longo prazo. O modelo LSTM bidirecional e parâmetros utilizado nesta pesquisa, retirados do trabalho [Stemle and Onysko 2018], foram implementados usando a biblioteca Keras[1] em Python 3.

O detector de metáforas LSTM é alimentado na camada de entrada A por uma sentença pré processada pela remoção das *stop words*. Cada palavra na sequência de texto é representada por um vetor contínuo de 100 *features*. Vetores contínuos, ou *word embeddings*, trata-se de uma forma de mapear palavras de um vocabulário para um espaço de dimensões reduzido em comum. Esta representação permite que o conhecimento sobre relações existentes entre as palavras, como similaridades semânticas e gramaticais, seja reutilizado em diferentes tarefas de NLP. No experimento foram utilizados vetores treinados através de rede neural fastText[2] no *dataset* BNC, contendo 100 milhões de tokens.

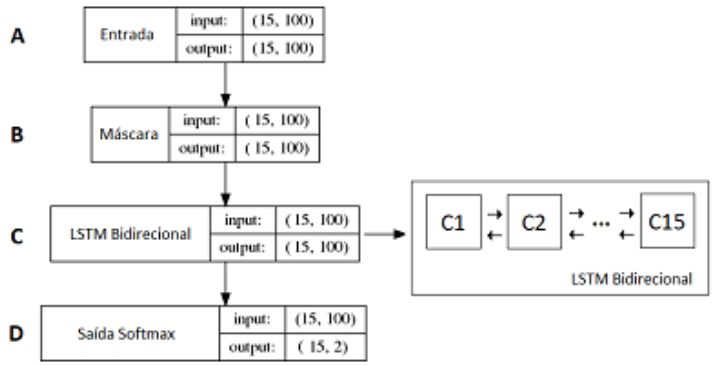


Figura 1. Topologia detector de metáforas.

Parâmetro	Valor
Tamanho da sequência	15
Features por palavra	100
Partições	3
Tamanho por batelada	32
Épocas de treinamento	20
Taxa de dropout	0.25
Word Embedding	BNC
Arquitetura	LSTM
Saída	*Softmax*

Figura 2. Parâmetros.

Na camada B sentenças menores que 15 palavras são completas por vetores nulos, enquanto sequências que ultrapassam o comprimento delimitado são separadas e analisadas separadamente. A camada seguinte é composta pelas células C1 à C15 do tipo LSTM, cada uma conectada ao vizinho anterior e sucessor, sendo responsável por processar a probabilidade da palavra ser uma metáfora dependendo de suas características capturadas pelo *word embeddings* e pelo estímulo recebido das palavras vizinhas.

4. Experimento

Disponibilizado de forma aberta[3], o *dataset* de notícias falsas é uma coleção de diferentes categorias de notícias extraídas de 745 sites de notícias (30GB de conteúdo). Além das classes confiável e falsa, o corpus inclui ainda categorias como sátira, *clickbait* e notícia de ódio. Para o treinamento foi utilizado o *dataset* padrão de metáforas da competição citada na Seção 2.

O experimento inicial realizado considerou somente os títulos das notícias, com o objetivo de verificar se estes seriam descritores suficientemente bons para a separação

[1]https://keras.io/
[2]https://embeddings.sketchengine.co.uk/static/index.html
[3]http://github.com/several27/FakeNewsCorpus/

das classe de notícias analisadas. Os resultados foram obtidos ao analisar uma amostra aleatória de 40000 títulos (aproximadamente 4MB). A execução do modelo sobre a amostra demandou 45 minutos de processamento em um processador Intel Core i3 (2Ghz) e 8GB de memória RAM. Os títulos possuem em média 8.2 palavras e 0.7 palavras marcadas como de uso metafórico.

Mesmo ao usar uma amostra pequena do *dataset*, foram observadas certas tendências como a da classe rumores apresentar maior uso de metáforas (média 1.53, p-value < 0.01) quando comparada à classe de notícias confiáveis (média 0.65). Na comparação das notícias falsas (média 0.73) com as notícias confiáveis, a diferença foi menor mas ainda estatisticamente relevante (p-value < 0.01).

Esse resultado pode ser um indicativo de que títulos de notícias falsas tendem a simular características como estilo linguístico empregado em títulos de notícias confiáveis, com o objetivo de enganar leitores ao esconder o viés presente no corpo da notícia.

5. Considerações Finais

Neste trabalho, procuramos estudar a utilização de metáforas na composição de textos do domínio jornalístico através de um modelo computacional conexionista. Em relação às classes gramaticais de metáforas, decidimos restringir a análise preliminar para verbos e substantivos, já que estas possuem maior frequência nas metáforas do corpus de treinamento. No entanto, o uso de metáforas de classes gramaticais menos frequentes poderia melhorar a classificação de notícias específicas.

Nos próximos passos do trabalho esperamos que a execução das análises nos corpos das notícias possa trazer resultados mais significativos, como padrões de usos de metáforas e de suas classes gramaticais característicos para os diferentes tipos de notícias ou para diferentes fontes. Tais distinções abririam espaço para análises mais aprofundadas, como por exemplo a verificação da existência de tipos de notícias similares em relação ao uso de metáforas, detecção da intenção do autor de acordo com o tipo da metáfora e correlações entre uso de metáforas e análise de sentimento no texto.

Referências

H. Thibodeau, P. and Boroditsky, L. (2011). Metaphors we think with: The role of metaphor in reasoning. *PLoS ONE*, 6(2):11.

Horne, B. D. and Adali, S. (2017). This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In *Eleventh International AAAI Conference on Web and Social Media*.

Lazer, D. M., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., Metzger, M. J., Nyhan, B., Pennycook, G., Rothschild, D., et al. (2018). The science of fake news. *Science*, 359(6380):1094–1096.

Lipton, Z. C. (2015). A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019.

Stemle, E. and Onysko, A. (2018). Using language learner data for metaphor detection. In *Proceedings of the Workshop on Figurative Language Processing*, pages 133–138.

Wee Leong, C., Beigman Klebanov, B., and Shutova, E. (2018). A report on the 2018 vua metaphor detection shared task. Technical report.

Proposta de uma arquitetura de Data Warehouse para análise de SDN e aplicações de Aprendizado de Máquina

Fernando Luiz Moro, Rodrigo Nogueira, Alexandre Amaral, Ana Paula Amaral

Instituto Federal de Educação, Ciência e Tecnologia Catarinense – Campus Camboriú
Caixa Postal 2016 – 88.340-055 – Camboriú – SC – Brasil

fernandoluizmoro@gmail.com, {rodrigo.nogueira, alexandre.amaral, ana.amaral}@ifc.edu.br

Abstract. *This paper presents the proposal of a data warehouse architecture that has as data source and object of study the IP flows and attacks in SDN. The proposal aims to provide a consistent and clean dataset for machine learning applications and any application that wishes to consume data of this feature. For the proposed objectives, a multidimensional database was developed, which is fed by an ETL stage based on the collection of network flows. Among the obtained results is the architecture itself, in which a dataset can be explored through OLAP queries by machine learning applications.*

Resumo. *Este artigo apresenta a proposta de uma arquitetura de data warehouse que tem como fonte de dados e objeto de estudo os fluxos IP e ataques em SDN. A proposta tem como objetivo fornecer um conjunto de dados consistente e limpo para aplicações de aprendizado de máquina e qualquer aplicação que deseje consumir dados desta característica. Para atingir os objetivos propostos, foi desenvolvido um banco de dados multidimensional, que é alimentado por uma etapa de ETL baseada na coleta de fluxos de rede. Dentre resultados obtidos é a arquitetura em si, na qual um conjunto de dados pode ser explorado através de consultas OLAP pelas aplicações de aprendizado de máquina.*

1. Introdução

Uma previsão realizada pela Forrester estimou que 500.000 dispositivos de *IoT (Internet of Things)* seriam comprometidos em 2017 [Moro 2017 *apud* Francis 2017]. Com o objetivo de prevenir e combater os ataques gerados por tais vulnerabilidades, tem sido empregado a integração entre as redes definidas por software (*Software-Defined Networking – SDN*) e as técnicas de aprendizado de máquina.

As redes definidas por software permitem através de um controle centralizado e homogêneo da rede o gerenciamento, a execução de tarefas de detecção e bloqueio de ataques de forma simplificada [Moro 2017 *apud* Ahmad *et al.* 2015]. Dentre as abordagens atuais aplicadas para a detecção de ataques em SDN, se destaca o emprego do aprendizado de máquina que vão além dos tradicionais métodos entrópicos que detectam uma anomalia já em andamento.

Os métodos de aprendizado de máquina permitem descobrir o ataque em uma rede, antes mesmo que este aconteça [Huang 2017]. No entanto, o grande desafio no emprego do aprendizado de máquina é que 80% de todo o esforço computacional é gasto na etapa de pré-processamento de dados [Losarwar 2012]. Um ambiente de *data*

warehouse, por sua vez, permite com que as dimensões sejam exploradas, já com os dados coletados, consistentes e limpos [Nogueira 2017]. Deste modo, quando aplicado os métodos de aprendizado de máquina, estes apenas se designam as suas reais tarefas.

2. Trabalhos relacionados

Com o grande volume de informações geradas, uma das principais causas para o crescimento do número de ataques está nas vulnerabilidades presentes nas atuais tecnologias. Isto mostra que os mecanismos para detectar e bloquear os ataques de redes se fazem necessários. Todavia, as atuais redes de computadores se tornaram complexas e heterogêneas, contendo dispositivos e softwares de inúmeros fabricantes com diferentes tecnologias e interfaces de acesso [Costa 2013].

[Huang 2017] desenvolveu um sistema de identificação de aplicativos que pode ser integrado com um sistema de gerenciamento de *QoS (Quality of Service)* em uma SDN. Em experimentos que resultaram em uma f-medida de 93.48%, o conjunto de dados continha diversas aplicações de teste. [Lopez 2017], ilustra em seu trabalho a avaliação de diversos métodos de aprendizado de máquina e a aplicação do algoritmo *PCA (Principal Component Analysis)*. O principal objetivo é realizar a seleção de atributos em cenários de análise de tráfego, no qual, o melhor resultado foi com seis características em árvores de decisão (97.4%) e o pior resultado foi com sete características em SVM-RFE (80.2%).

Exemplo da integração entre técnicas de *data warehousing* e aprendizado de máquina, é o caso de [Mansmann 2014], que obteve um modelo multidimensional da rede social *Twitter* e desenvolveu um ambiente de *data warehouse* que permitiu a criação de um cubo de dados, bem como a análise de sentimentos. [Nogueira 2017], em uma abordagem similar, desenvolveu um ambiente de *data warehouse* que coleta notícias em tempo real com um algoritmo de aprendizado de máquina que realiza o enriquecimento semântico na etapa de ETL (*Extract, Transform, Load*). O mesmo *data warehouse*, serve como fonte de dados para diversas aplicações através de uma API REST (*Representational State Transfer Application Programming Interface*).

Tomando conhecimento das abordagens da literatura este trabalho foi construído baseado na seguinte hipótese: “É possível desenvolver um *data warehouse* baseado em uma rede definida por software para alimentar as aplicações de aprendizado de máquina?”.

3. Arquitetura proposta de um data warehouse para análise de SDN

A arquitetura proposta neste trabalho está ilustrada na Figura 1. A fonte de dados é obtida através da coleta dos fluxos IP seguindo a metodologia proposta por [Amaral 2015]. Uma vez coletados, é realizada a etapa de limpeza e transformação dos dados, na qual destaca-se principalmente a transformação das datas para o padrão do modelo multidimensional. Posteriormente, é realizada a carga no banco de dados multidimensional, a partir do qual é possível a exploração do cubo de dados através de consultas OLAP (*Online Analytical Processing*). A implementação segue uma arquitetura HOLAP (Hybrid *Online Analytical Processing)* utilizando o servidor PostgreSQL 10.

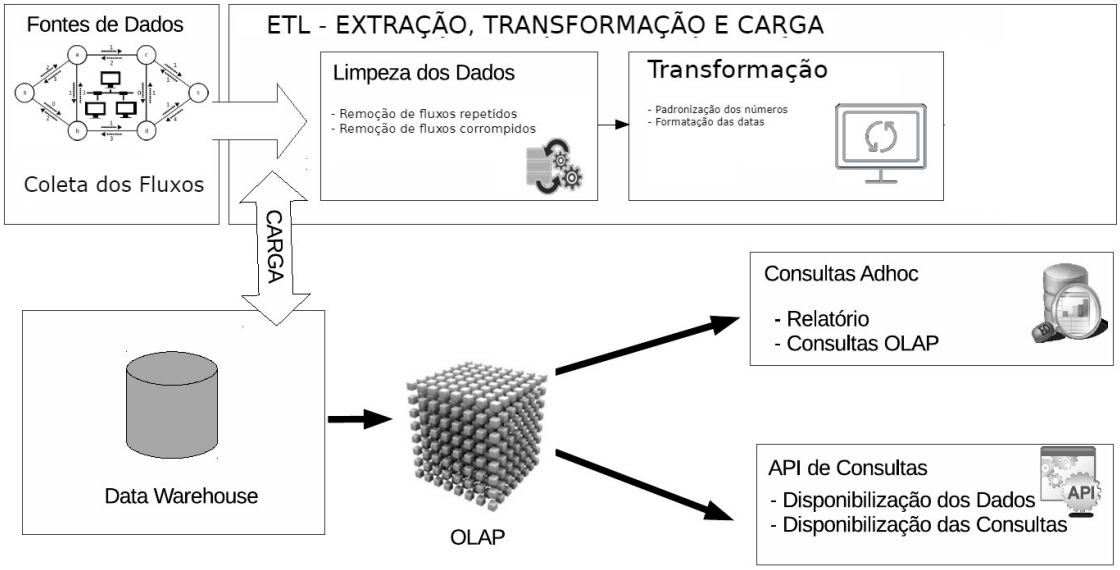


Figura 1. Arquitetura do data warehouse proposta

O banco de dados multidimensional é responsável por consolidar e armazenar os fluxos IP coletados e pré-processados. Para tal, foi utilizado o modelo de estrela proposto por [Kimball 2011] conforme é mostrado na Figura 2. No modelo desenvolvido, o objeto de análise é o fluxo IP, no qual as dimensões fornecem métricas para avaliar o comportamento da rede, principalmente, em cenários de ataque.

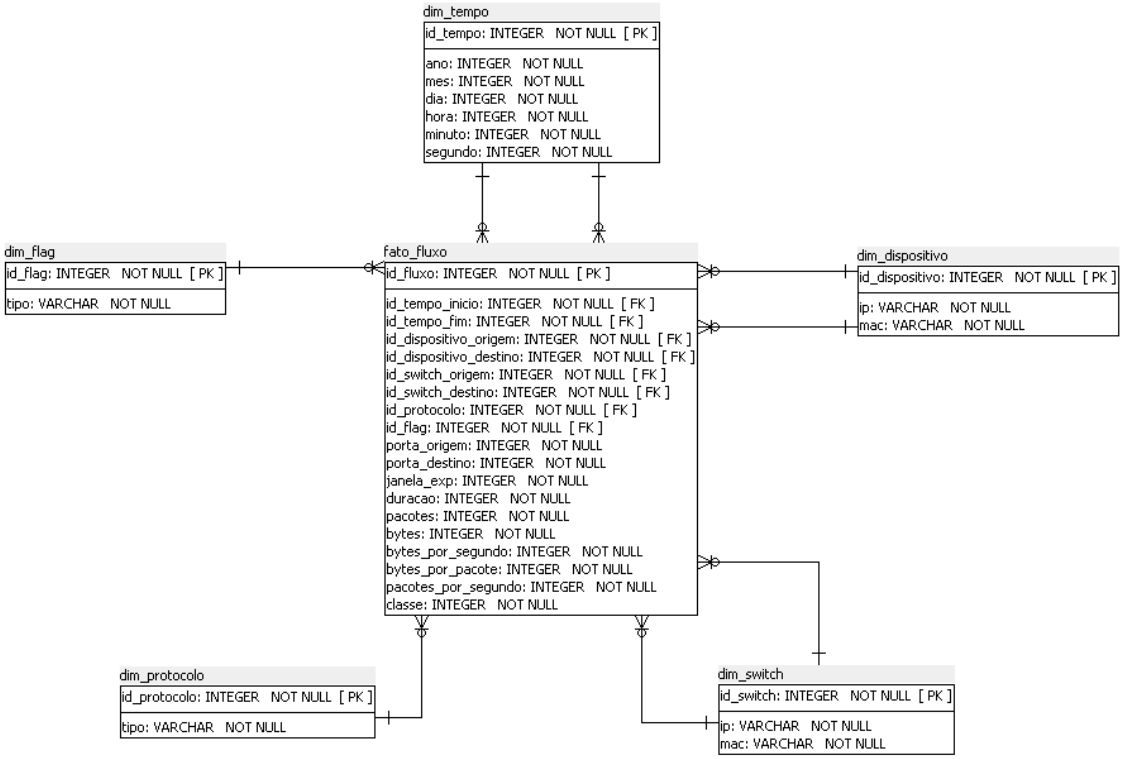


Figura 2. Modelo multidimensional da arquitetura proposta

4. Considerações finais, resultados obtidos e esperados

A partir da arquitetura desenvolvida é possível realizar a exploração do cubo de dados respondendo questões como: “Qual é o IP que mais atacou?”, “Qual a média de ataques?”, “Qual o período que mais houve um ataque?”. Um cubo de dados é amplo, e espera-se que através de sua exploração seja possível realizar experimentos e avaliar o desempenho da arquitetura desenvolvida no emprego de algoritmos de aprendizado de máquina.

Este artigo é fruto de uma pesquisa interdisciplinar em andamento, que integra as áreas de redes de computadores, segurança da informação, banco de dados e inteligência artificial. Deste modo, o que foi apresentado até o momento está em constante desenvolvimento e os experimentos aqui citados consistem em trabalhos futuros.

Referências

Amaral, A. A. (2015). Computação autonômica aplicada ao diagnóstico e solução de anomalias de redes de computadores. Universidade Estadual de Campinas (UNICAMP).

Costa, L. R. (2013). OpenFlow e o Paradigma de Redes Definidas por Software. Universidade de Brasília.

Huang, N.-F., Li, C.-C., Li, C.-H., et al. (2017). Application identification system for SDN QoS based on machine learning and DNS responses. In *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE.

Kimball, R. and Ross, M. (2011). *The data warehouse toolkit: the complete guide to dimensional modeling*. 2nd revised ed. Canada: John Wiley and Sons, Inc.

Lopez, M. A., Lobato, A. G. P., Mattos, D. M. F. and Alvarenga, I. D. (2017). Um Algoritmo Não Supervisionado e Rápido para Seleção de Características em Classificação de Tráfego. In *XXXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. Sociedade Brasileira de Computação (SBC).

Losarwar, V. and Joshi, D. M. (2012). Data Preprocessing in Web Usage Mining. In *International Conference on Artificial Intelligence and Embedded Systems (ICAIES’2012)*.

Mansmann, S., Ur Rehman, N., Weiler, A. and Scholl, M. H. (2014). Discovering OLAP dimensions in semi-structured data. *Information Systems*, v. 44, p. 120–133.

Moro, F. L., Amaral, A., Amaral, A. P. and Nogueira, R. (nov 2017). Detecção e autorreparo de anomalias em redes definidas por software. In *XVII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*. Sociedade Brasileira de Computação (SBC).

Nogueira, R. (2017). Newsminer: um sistema de data warehouse baseado em textos de notícias. Universidade Federal de São Carlos (UFSCar).

Desenvolvimento de um sistema para a classificação de *Fakenews* com Textos de Notícias em língua Portuguesa

Roger Oliveira Monteiro, Rodrigo Ramos Nogueira, Greisse Moser

Centro Universitário Leonardo da Vinci – UNIASSELVI - BR 470 - Km 71

roger.o.monteiro@gmail.com,rodrigo.nogueira@uniasselvi.com.br,

greisse.moser@uniasselvi.com.br

Resumo. *Com o rápido avanço da tecnologia e o fácil acesso e disseminação de informações, o termo fakenews vem ganhando preocupante atenção e pesquisas em diversas áreas vêm sendo desenvolvidas. Sendo assim, o objetivo deste trabalho é usar métodos de aprendizado de máquina para descobrir, classificar e armazenar textos de notícias falsas, para posterior aplicação a etapa ETL de um Data Warehouse e um ambiente de consulta que contribuirá com pesquisas futuras. Para isso foi criado um dataset e os métodos Regressão Logística, Naive Bayes e SVM foram avaliados. Finalizando o trabalho com a seleção do melhor método que foi inserido em um sistema de avaliação online de notícias falsas.*

1. Introdução

Diante da facilidade com que hoje em dia qualquer pessoa pode ter acesso a informação, e com a facilidade do seu uso, vivenciamos uma era de grandes avanços e soluções, seguido porém, por problemas ainda maiores, como é o caso das notícias falsas. Segundo MONTEIRO et al. (2018), devido à sua natureza atraente, as notícias falsas se espalham rapidamente, influenciando o comportamento das pessoas em diversos assuntos, desde questões saudáveis (por exemplo, revelando medicamentos milagrosos) até política e economia (como no recente escândalo Cambridge Analytica / Facebook e na situação Brexit).

Dado seu destaque, tem sido realizadas diversas multidisciplinares sobre o tema. Almejando contribuir com tais pesquisas, este trabalho tem como objetivo acoplar à etapa de ETL (*Extract, Transform, Load*) de um *Data Warehouse* de Notícias o enriquecimento semântico através de classificação do tipo de notícias: real ou falsa.

2. Trabalhos Correlatos

No que se refere à notícias falsas e a aplicação de *Machine Learning*, GRUPPI et al. (2018) construíram um dataset com notícias, em português e inglês, tendo por objetivo construir um classificador para predizer se a fonte da notícia é ou não confiável. Rodando um algoritmo de SVM com um kernel linear, foi possível estabelecer as características mais importantes, bem como sua classificação. Como resultado, o algoritmo de classificação obteve acurácia de 85% para os datasets brasileiros e 72% para datasets Americanos.

Em uma contribuição para a área de classificação de notícias, MONTEIRO et al. (2018) utilizam o dataset Fake.br com o objetivo de avaliar os principais métodos de

pré-processamento de textos para avaliar o desempenho do método SVM. Os melhores resultados foram obtidos com a combinação de *bag-of-words* com sentimentos, bem como o uso de todos os atributos, ambos com acurácia de 90%.

MARUMO (2018) coletou notícias de sites com notícias verídicas e sites com notícias falsa e/ou de cunho satírico, com o objetivo de encontrar o melhor método para detecção de fakenews. Como parte do pré processamento dos dados, utilizou-se o framework Gensim para remoção de caracteres não alfabéticos, a substituição de espaçamentos e quebra de linhas para espaços únicos, remoção de palavras com menos de 3 caracteres e a conversão de letras maiúsculas para minúsculas. Também foi utilizado o framework keras para tokenização dos dados. Com a aplicação dos algoritmos de classificação LSTM e SVM, conseguiu-se uma acurácia acima de 90%.

No que se refere ao enriquecimento semântico em ambientes de Data Warehouse através do emprego de técnicas de *Machine Learning*, é o caso Mansman (2014), que obteve um modelo multidimensional da rede social Twitter e desenvolveu um ambiente de Data Warehouse que permitiu a criação de um cubo de dados, bem como a análise de sentimentos. Nogueira (2018), em uma abordagem similar, desenvolveu um ambiente de Data Warehouse que coleta notícias em inglês em tempo real, no qual após avaliação regressão logística, Naïve Bayes, SVM e Perceptron tiveram resultados próximos, dos quais o este último foi utilizado para realizar o enriquecimento semântico na etapa de ETL.

3. Metodologia - Proposta de Aplicação

Após pesquisas por base de dados com *fakenews*, verificamos que existem poucos recursos disponíveis no idioma Português do Brasil, no qual o dataset mais utilizado é o Fake.br (MONTEIRO et al., 2018). A proposta apresentada, tem como objetivo proporcionar um ambiente com dados consistentes e limpos na forma de um corpus multidimensional para consumo por aplicações externas e usuários. O corpus multidimensional é um conjunto de textos armazenados de acordo com um modelo multidimensional, que permite explorar a multidimensionalidade em diferentes níveis de abstração: tempo, categoria das notícias, tipo (verdadeira ou *fakenews*).

A metodologia deste trabalho é baseada na arquitetura proposta por NOGUEIRA(2018), na qual o classificador gerado será acoplado a etapa de ETL de um Data Warehouse gerando o enriquecimento semântico em uma nova dimensão.

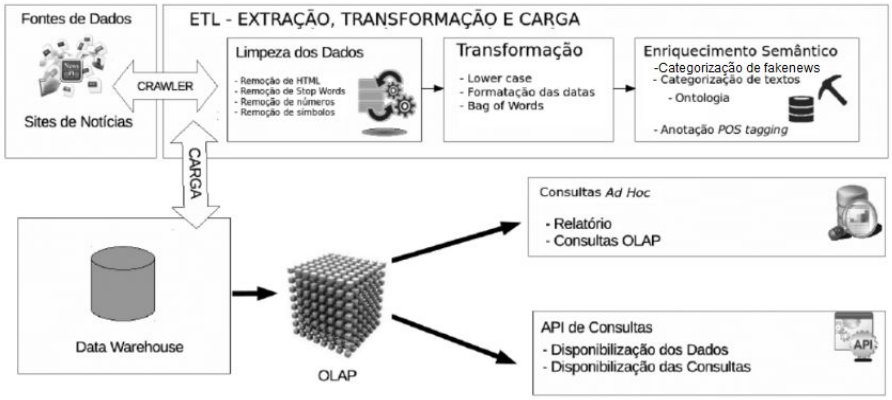


Figura 1. Arquitetura utilizada, adaptada de Nogueira (2018).

Para realizar os experimentos foi desenvolvido um web crawler, utilizando a linguagem python, juntamente com a biblioteca *beautiful soup*, *chromium web driver* e *selenium web driver*, para a coleta inicial dos dados. Foi construído um dataset composto por 2714 títulos de notícias falsas coletadas do site *boatos.org* e 3185 títulos de notícias verdadeiras coletadas do site *brasil.elpais.com*. Inicialmente será utilizado apenas os títulos das notícias. Posteriormente, planejamos a utilização da notícia por inteiro.

A partir da criação de um sistema de coleta, com um algoritmo acoplado à etapa de ETL, este irá automaticamente classificar os dados coletados, aumentando assim a acurácia do classificador, e gerando uma base maior de dados para futuros trabalhos de combate a *fakenews*. Também foi construído uma interface *Web*, onde o usuário será capaz de submeter um link e verificar se este é ou não uma notícia verdadeira, servindo este como protótipo antes de ser submetido a etapa de ETL(sendo esta, o propósito geral deste trabalho).

	titulo_noticia	url	label
0	Indústria brasileira rea ...	https://brasil.elpais.com/brasil/20:...	0
1	A bancarrota de Detro:...	https://brasil.elpais.com/brasil/20:...	0
2	PIB no Brasil cai 0...	https://brasil.elpais.com/brasil/20:...	0
3	O órgão supervisor eur:...	https://brasil.elpais.com/brasil/20:...	0
4	Vega S...	https://brasil.elpais.com/brasil/20:...	0

	titulo_noticia	url	label
0	Neto de Chico Buarque f...	www.boatos.org/entretenim...	1
1	Correios, em 2018, c...	www.boatos.org/tecnologia...	1
2	Caseiro do sítio de ,...	www.boatos.org/politica...	1
3	Vídeo mostra rato tomanc...	www.boatos.org/mundo/video....	1
4	Lutadora de vale tudo:...	www.boatos.org/esporte/lutr...	1

Tabela 1. Cinco primeiras linhas de ambos datasets.

Posteriormente, utilizando a literatura como referência foram selecionados três métodos para serem avaliados no dataset: Regressão Logística (Logistic Regression), Naive Bayes e SVM. Após a avaliação o melhor método será acoplado à etapa de ETL do sistema proposto, bem como a interface Web de classificação de notícias.

4. Resultados Parciais

Os dados obtidos receberam tratamento de valores nulos, ruídos (caracteres especiais, tais como vírgulas, pontos, parênteses, etc) e transformação para letras minúsculas. Cada dataset recebeu uma nova coluna, chamada label, onde foi atribuído o valor *booleano* 0 para notícias verdadeiras, e 1 para as notícias falsas. Com isso, os dados foram combinados em um único dataset. Os rótulos das colunas foram convertidos em valores numéricos utilizando o Label Encoder do pacote scikit-learn.

O dataset foi então dividido entre treino e teste, na proporção de 75% e 25% respectivamente. A primeira parcela serve para treinar o algoritmo, enquanto a segunda, para verificar a acurácia do mesmo. Na sequência, receberam tratamento de tokenização, utilizando o pacote NLTK, com o *bag of words* em português do Brasil.

Testes efetuados utilizando os algoritmos Regressão Logística (Logistic Regression), Naive Bayes e SVM (kernel linear), obtiveram a acurácia de 90.33%, 89.27% e 90.52% respectivamente, no modelo de testes. Os resultados parciais obtidos após a construção, treino e produção do modelo foram satisfatórios. O algoritmo escolhido para a implementação inicial foi o SVM, que além de obter o melhor desempenho, mostrou-se bastante recorrente na literatura consultada. Como técnica de

avaliação do modelo empregado, foi utilizado a validação cruzada com o método k-fold = 10.

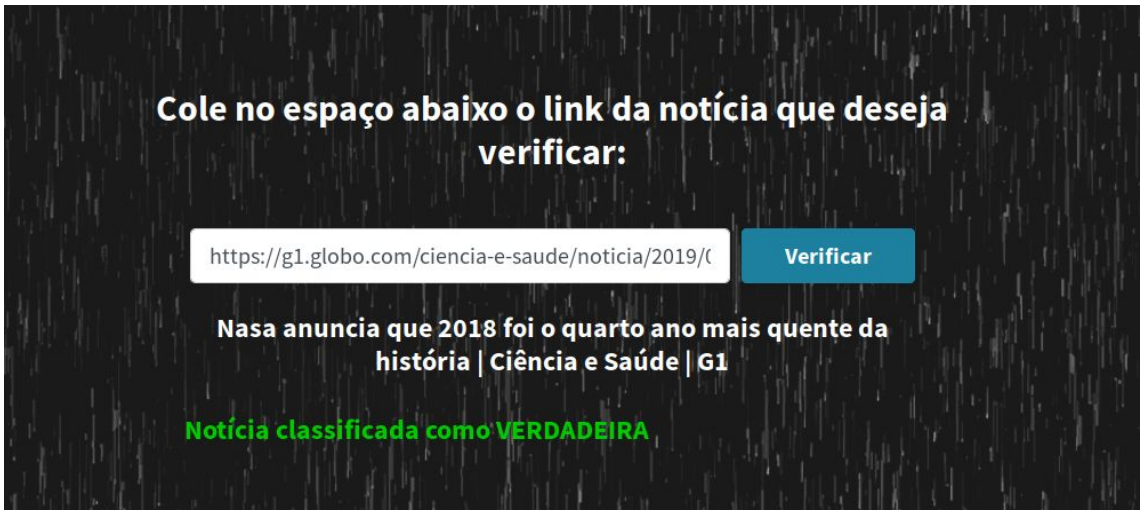


Figura 2. Interface Web da Aplicação desenvolvida. Disponível em: <https://detectorfakenews.herokuapp.com/>. Acesso em 18 fev. 2018.

5. Considerações Finais e Trabalhos Futuros

O estudo mostrou-se relevante para o aperfeiçoamento e entendimento dos envolvidos, bem como a corroboração da necessidade do combate às *fake news.* Para futuros trabalhos, tem-se como objetivo avaliar outras características técnicas de pré-processamento, aumentar a base de treino, utilizar além do título, a notícia por completo, aplicar os novos resultados a interface *web,* e posteriormente, o acoplamento a ETL do *Data Warehouse.*

Referências

GRUPPI, Maurício; HORNE, Benjamin D.; ADALI, Sibel. “An Exploration of Unreliable News Classification in Brazil and The U.S.” Rensselaer Polytechnic Institute, Troy, New York, USA.2018.

MANSMANN, Svetlana; REHMAN, Nafees Ur; WEILER, Andreas; SCHOLL, Marc H. “Discovering OLAP dimensions in semi-structured data.” Information Systems, v. 44, p. 120-133, 2014.

MARUMO, Fabiano Shiiti. “Deep Learning para classificação de Fake News por sumarização de texto.” - Londrina, 2018.

MONTEIRO, Rafael A.; SANTOS, Roney L. S.; PARDO, Thiago A. S.; ALMEIDA, Tiago A. de; RUIZ, Evandro E. S.; VALE, Oto A.. “Contributions to the Study of Fake News in Portuguese: New Corpus and Automatic Detection Results.” In: International Conference on Computational Processing of the Portuguese Language. Springer, Cham, 2018. p. 324-334.

NOGUEIRA, Rodrigo Ramos. O Poder do Data Warehouse em Aplicações ed Machine Learning: Newsminer: Um Data Warehouse Baseado em Textos de Notícias. São Paulo: Nea, 2018.

Integração semântica entre dados dos domínios da educação e segurança: um caso de Curitiba

Pedro Henrique Stolarski Auceli[1], Rita C. G. Berardi[1] Nadia P. Kozievitch[1]

Departamento Acadêmico de Informática
Universidade Tecnológica Federal do Paraná (UTFPR) – Curitiba, PR – Brazil
pedroauceli@gmail.com, ritaberardi@utfpr.edu.br, nadiap@utfpr.edu.br

Abstract. *The objective of this work is to analyze if there is a relationship between the students' income and the number of police occurrences in the neighborhood in which they are based on the semantic integration of heterogeneous open databases of education and security domains. To perform the integration an ontology is proposed with the intention of semantically unify the base domain and formally specify the data relationship.*

Resumo. *O objetivo desse trabalho é analisar se existe uma relação entre o rendimento de alunos com a quantidade de ocorrências policiais do bairro em que se encontram a partir da integração semântica de bases de dados abertas heterogêneas dos domínios de educação e segurança. Os dados utilizados são referentes à cidade de Curitiba-Paraná. Para realizar a integração uma ontologia é proposta com o intuito de unificar semanticamente o domínio das bases e especificar formalmente o relacionamento dos dados.*

1. Introdução

Atualmente no Brasil existem várias bases de dados abertas e, pelo fato de cada uma ser de um domínio específico diferente, realizar uma integração entre elas para recuperar informações úteis e mais complexas é uma tarefa não trivial. Isso se deve às diferentes semânticas dos dados, ou seja, a diferença do significado dos dados nas diferentes bases, que são modeladas, coletadas e abertas de maneira independente.

O objetivo desse trabalho é a integração semântica de bases de dados, assim possibilitando novos tipos de análises sobre os dados. Para alcançar tal objetivo é criada uma ontologia, que é uma especificação de uma realidade (GUARINO; OBERLE; STAAB, 2009), que pode ser utilizada para integrar bases de dados de domínios diferentes, uma vez que nela será especificado um novo domínio que unifica as bases. No caso deste trabalho as bases de dados são da área da educação e da segurança pública. As bases foram escolhidas a partir do argumento de Junior et al. (2018), que diz que trabalhos com dados abertos governamentais devem ser feitos com dados que sejam relevantes para a população, com o intuito de melhorar os serviços ofertados pelo governo. O método apresentado por Pereira, Salvador, Wassermann (2018) será utilizado para avaliar a ontologia. Esse consiste em criar uma pergunta que deve ser respondida com informações apenas obtidas através dos dados integrados (PEREIRA, SALVADOR, WASSERMANN, 2018). No caso deste trabalho a pergunta feita é: existe uma relação entre o rendimento e as notas de escolas com a quantidade de ocorrências policiais do bairro em que se encontram?

2. Bases de dados

As bases de dados que são utilizadas neste trabalho são referentes à cidade de Curitiba: SiGesGuarda e Unidades de Atendimento de Curitiba ativas, além dos datasets de Média e Rendimento dos alunos por região. Tanto a base de nota média de escolas quanto a de rendimentos foram obtidas através do portal de dados abertos do governo brasileiro[1], e contêm respectivamente as notas de acordo com as turmas de cada escola dentro do país e o rendimento das mesmas. O rendimento é um cálculo feito através da taxa de aprovação, reprovação e abandono dos alunos. Ambas as bases se encontram no formato xls (formato proprietário do Microsoft Excel) e podem ser convertidas para csv (*Comma-separated Values*). A base da SiGesGuarda é referente aos dados de atendimentos feitos pela guarda municipal da cidade de Curitiba, que pode ser obtida em formato csv através do portal de dados abertos da cidade de Curitiba[2]. A base de unidades de atendimento ativas também é disponibilizada através do portal de dados abertos de Curitiba, e é referente às unidades de atendimento de uso público.

3. Metodologia

O primeiro passo para conseguir integrar as bases foi fazer a limpeza, a normalização e redução da granularidade dos dados de cada uma delas. O tratamento dos dados é necessário para facilitar a comparação, e para poder inserir os dados dentro de um banco de dados relacional. Para utilizar o plugin Ontop, utilizado no framework Ontop apresentado por Pereira, Salvador, Wassermann (2018), o banco de dados relacional é necessário, pois ele é o responsável pela distribuição dos dados que servirão para povoar a ontologia. Enquanto que a redução foi feita para minimizar o custo computacional e diminuir a complexidade da especificação da ontologia. O passo seguinte foi a inserção dos dados obtidos através das bases de dados no banco de dados PostgreSQL. Vale ressaltar que foram utilizados apenas 500 registros da base da SiGesGuarda, e que os dados das outras 3 bases foram inseridos manualmente pelo autor, em uma quantidade suficiente para testar a ontologia. Isso ocorreu devido à grande quantidade de ruídos que dificultaram o trabalho de limpeza e inserção dos dados.

Utilizando a ferramenta Protégé[3], foi criada a ontologia (Figura 2) com suas classes, relacionamentos e propriedades necessárias para responder à questão de competência motivadora ao experimento. No total foram criadas 3 classes: Bairro, GuardaMunicipal e Escola. 1 relacionamento: hasBairro, que liga um registro da classe GuardaMunicipal ou Escola com um bairro. E 8 propriedades de dados: nomeEscola, nomeBairro que são do tipo *String,* mesRegistro, anoRegistro, codigoRegistro, codigoBairro que são do tipo *int* e mediaEscola, rendimentoEscola que são do tipo *float*.

Com a ontologia devidamente criada foi necessário definir as regras para o mapeamento do banco de dados relacional. Na Figura 1 são apresentados todos os mapeamentos que foram necessários entre a ontologia e o banco de dados relacional.

[1] Governo Brasileiro. Portal brasileiro de dados abertos.< http://dados.gov.br/group/educacao>

[2] Prefeitura de Curitiba. Portal de dados abertos da cidade de Curitiba. <http://www.curitiba.pr.gov.br/dadosabertos/consulta/>

[3] The Protégé project: <https://protege.stanford.edu/>

4. Resultados

Com a ontologia povoada a integração foi obtida e sua avaliação pode ser realizada. Para avaliar se a ontologia foi suficiente para a integração, será utilizada a questão de competência apresentada na introdução. Para isso foi utilizado outro plugin chamado Ontop SPARQL, que permite a criação de *queries* em SPARQL que serão executadas sobre a ontologia.

```
urn:MAPID-12eb4cd3f1534914aa2d650fed237528
:GuardaMunicipal{codigo} a :GuardaMunicipal ; :mesRegistro {mes} ; :anoRegistro {ano} ; :codigoRegistro {codigo} ; :hasBairro :{bairro} .
select codigo, mes,ano,bairro from registro

urn:MAPID-96bf884682384951984289cdd87cd201
:{nomeLocal} a :Escola ; :nomeEscola {nomeLocal} ; :mediaEscola {mediaEscola} ; :rendimentoEscola {rendimento} ; :hasBairro :{nomebairro} .
select estabelecimento.nomeLocal, mediaEscola, rendimento, nomebairro from escolar, escolam,estabelecimento where (escolar.nomeEscola =
escolam.nomeEscola and escolar.codigo=estabelecimento.codigoLocal)

urn:MAPID-4e3395fb9ee84c4493ab5b6b49604e0f
:{nomeBairro} a :Bairro ; :nomeBairro {nomeBairro} ; :codigoBairro {codigoBairro} .
select nomeBairro, codigoBairro from estabelecimento
```

Figura 1. Mapeamentos entre a ontologia e o banco de dados relacional

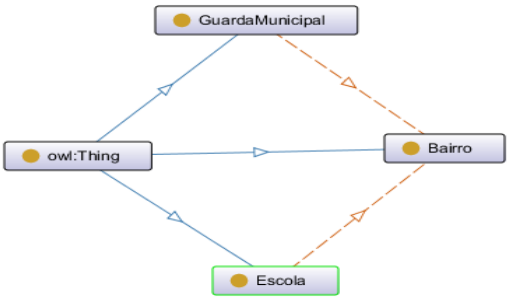


Figura 2. Ontologia

Uma dificuldade encontrada foi o fato da função *GROUP BY* não ter sido implementada pela equipe do Ontop, logo não foi possível utilizar a função *count* e não foi possível encontrar o Bairro com a maior quantidade de registros de atendimentos. Para resolver isso foram feitas *queries* específicas para cada bairro como pode ser visto na query abaixo, em que é tratado especificamente o bairro Bacacheri da cidade de Curitiba.

```
PREFIX tes: http://example.org/
SELECT ?registro ?nome ?escola ?media ?rendimento
WHERE {
?bairro tes:nomeBairro ?nome.
?bairro tes:nomeBairro "bacacheri".
?registro tes:hasBairro ?bairro.
?registro a tes:GuardaMunicipal.
?escola tes:hasBairro ?bairro.
?escola a tes:Escola.
?escola tes:mediaEscola ?media.
?escola tes:rendimentoEscola ?rendimento
}
```

rendimento	escola	nome	media	registro
"98.0"^^xsd:double	<http://example.org/Rosario>	"bacacheri"	"92.0"^^xsd:double	<http://example.org/GuardaMuni...
"98.0"^^xsd:double	<http://example.org/Rosario>	"bacacheri"	"92.0"^^xsd:double	<http://example.org/GuardaMuni...
"98.0"^^xsd:double	<http://example.org/Rosario>	"bacacheri"	"92.0"^^xsd:double	<http://example.org/GuardaMuni...
"98.0"^^xsd:double	<http://example.org/Rosario>	"bacacheri"	"92.0"^^xsd:double	<http://example.org/GuardaMuni...
"98.0"^^xsd:double	<http://example.org/Rosario>	"bacacheri"	"92.0"^^xsd:double	<http://example.org/GuardaMuni...
"98.0"^^xsd:double	<http://example.org/Rosario>	"bacacheri"	"92.0"^^xsd:double	<http://example.org/GuardaMuni...
"98.0"^^xsd:double	<http://example.org/Rosario>	"bacacheri"	"92.0"^^xsd:double	<http://example.org/GuardaMuni...
"98.0"^^xsd:double	<http://example.org/Rosario>	"bacacheri"	"92.0"^^xsd:double	<http://example.org/GuardaMuni...
"98.0"^^xsd:double	<http://example.org/Rosario>	"bacacheri"	"92.0"^^xsd:double	<http://example.org/GuardaMuni...
"98.0"^^xsd:double	<http://example.org/Rosario>	"bacacheri"	"92.0"^^xsd:double	<http://example.org/GuardaMuni...
"98.0"^^xsd:double	<http://example.org/Rosario>	"bacacheri"	"92.0"^^xsd:double	<http://example.org/GuardaMuni...
"98.0"^^xsd:double	<http://example.org/Rosario>	"bacacheri"	"92.0"^^xsd:double	<http://example.org/GuardaMuni...
"98.0"^^xsd:double	<http://example.org/Rosario>	"bacacheri"	"92.0"^^xsd:double	<http://example.org/GuardaMuni...

Figura 3. Resultados da *query*

Na Figura 3 é apresentado o resultado da *query* com o bairro com a maior quantidade de atendimentos feitos pela guarda municipal. Onde nesse caso o bairro com a maior ocorrência foi o bairro "Bacacheri", e graças à integração é possível verificar a média e o rendimento das escolas da área. A Figura 3 deixa evidente as limitações da ferramenta, por não conseguir agrupar os registros, porém também mostra que é possível obter a integração e fazer uma análise dos dados.

5. Considerações finais

O experimento realizado neste trabalho teve resultados satisfatórios, uma vez que foi possível obter uma análise com base nos dados integrados. Porém vale ressaltar que a análise foi feita com uma porção dos dados disponíveis pelas bases de dados.

Foram observadas dificuldades com relação às bases de dados, uma vez que é necessária uma limpeza dos dados antes da utilização desses. Além disso, foram encontrados problemas na implementação, uma vez que algumas funções do plugin Ontop SPARQL, como a *GROUP BY* e a *count,* ainda não foram implementadas, o que dificultou a obtenção de resultados através da análise das *queries*. Outro problema encontrado foi a redundância no mapeamento de classes, principalmente para a classe "Bairro", onde novas instâncias eram criadas para cada registro das bases de dados que continham um bairro.A utilização da ontologia e dos mapeamentos entre a ontologia e os bancos de dados relacionais se mostrou promissora para a realização de uma integração entre os domínios heterogêneos. Nas próximas análises serão utilizados mais dados e novas perguntas serão executadas sobre os domínios integrados.

References

Guarino N., Oberle D., Staab S. (2009) "What Is an Ontology?". In: Staab S., Studer R. (eds) Handbook on Ontologies. International Handbooks on Information Systems. Springer, Berlin, Heidelberg

JUNIOR, F. T. M. et al. "Avaliação da prontidão para abertura de dados das instituições públicas brasileiras: caso de uma instituição financeira pública brasileira". Brazilian Journal of Information Studies: Research Trends, 2018.

Pereira, D. L. N. C., Wassermann, R., Salvador, L. Integração Semântica das Bases de Dados do Município de São Paulo: Um Estudo de Caso com Anomalias Congênitas. XI Seminar on Ontology Research in Brazil, ONTOBRAS 2018.

Framework Ontop. Disponível em: <https://ontop.inf.unibz.it/>

Musen, M.A. The Protégé project: A look back and a look forward. 2015. Association of Computing Machinery Specific Interest Group in Artificial Intelligence.

Visualização de dados do Índice de Qualidade da Água aplicado a múltiplos pontos em um Sistema de Informação Ambiental

Vania Elisabete Schneider[1], Odacir Deonisio Graciolli[2], Helena Graziottin Ribeiro[2], Adriano Gomes da Silva[1], Mayara Cechinatto[1]

[1]Instituto de Saneamento Ambiental – Universidade De Caxias do Sul (UCS)
Rua Francisco Getúlio Vargas, 1130 - 95070-560 - Caxias do Sul - RS – Brasil

[2]Universidade De Caxias do Sul (UCS)
Rua Francisco Getúlio Vargas, 1130 - 95070-560 - Caxias do Sul - RS - Brasil

{veschnei,odgracio,hgrib,agsilva11,mcechinatto}@ucs.br

Abstract. *Information Systems may be configured as tools to support the study and decision making related to environmental issues. Together with databases in a datawarehouse format, these decisions may be geared towards the historical scope of the data. The Environmental Information System - SIA, was developed to store and allow queries of environmental historical data from the Taquari-Antas Hydrographic Basin, in which are installed several hidroeletric plants. This paper presents the development of data visualization capabilities for the Water Quality Index at multiple points in the region.*

Resumo. *Sistemas de Informação podem se configurar como ferramentas para apoio ao estudo e à tomada de decisões relacionadas às questões ambientais. Em conjunto com bancos de dados em formato datawarehouse, estas decisões podem estar voltadas ao âmbito histórico dos dados. O Sistema de Informação Ambiental - SIA, foi desenvolvido para armazenar e permitir a consulta de dados históricos ambientais de diversas centrais hidrelétricas instaladas na Bacia Hidrográfica Taquari-Antas. Esse trabalho apresenta o desenvolvimento de recursos de visualização de dados para o Índice de Qualidade da Água em múltiplos pontos da região.*

1. Introdução

Informações sobre o meio ambiente tornaram-se mais necessárias e detalhadas na medida em que a sua preservação foi adquirindo importância como política pública ao redor do mundo [GUNTHER 1997]. Dentre os requerentes desse tipo de informação estão as empresas que precisam reportar o seu impacto ambiental aos órgãos de fiscalização. As Tecnologias da Informação (TI) se apresentam como grandes aliadas no armazenamento de dados históricos e no processo de tomada de decisão, uma vez que a informação precisa estar disponível para o gestor em grande escala e de forma condensada [O'BRIAN e MARAKAS 2007].

Para atender as necessidades de armazenamento histórico, consultas considerando diferentes granularidades e exposição aos órgãos de fiscalização as informações ambientais coletadas ao longo de anos por empreendimentos hidrelétricos

instalados na Bacia Taquari-Antas[1], localizada a nordeste do estado do Rio Grande do Sul, foi desenvolvido o Sistema de Informações Ambientais – SIA. Os dados utilizados pelo SIA são pertinentes à qualidade da água, ao clima e à fauna da região. Para permitir o armazenamento temporal e processamento analítico, esses dados estão armazenados em um Data Warehouse (DW), um banco de dados que armazena conjuntos de dados históricos de longos períodos para que estes possam ser processados e disponibilizados à gerência com diferentes níveis de detalhe para fornecer indicadores para análise [ELMASRI e NAVATHE 2018], com uma subdivisão em datamarts [KIMBALL e ROSS 2013] separados pelos domínios dos módulos, compartilhando da mesma dimensão tempo. Para tornar mais simples a compreensão dos dados, algo de suma importância para a ciência dos dados e suporte à tomada de decisão [CAO 2017], [MOORE 2017], [BIKAKIS 2018], são utilizados elementos de visualização das informações como relatórios, tabelas, gráficos e um webmapa, para a produção de indicadores, análises estatísticas, consultas a índices e comparações com determinadas legislações ambientais, permitindo a seleção de diferentes filtros de consulta, como agrupamento por regiões e período.

Dentre os índices presentes no sistema está o Índice de Qualidade da Água (IQA), o qual possui o objetivo de avaliar a qualidade da água bruta para sua disponibilização para o abastecimento público após o tratamento [VON SPERLING 2007]. Seus parâmetros são, em sua maioria, indicadores de contaminação causada pelo lançamento de esgotos domésticos [ANA 2018].

Utilizando os dados históricos de monitoramento de qualidade da água armazenados no DW do SIA, este trabalho tem por objetivo apresentar o desenvolvimento de recursos para a visualização agrupada do IQA de diferentes pontos de monitoramento presentes no módulo de qualidade da água. A necessidade de se desenvolver uma ferramenta através da qual o IQA possa ser estudado de forma agrupada e em diferentes pontos da bacia se deve à importância da visualização de dados de indicadores em modo comparativo para a tomada de decisões.

2. Metodologia

O SIA é uma aplicação acessível pela web e utiliza a estrutura cliente-servidor [SOMMERVILLE 2011]. No lado servidor o desenvolvimento foi na linguagem PHP. No lado cliente são utilizadas as linguagens HTML, CSS e de programação Javascript para processar as requisições e enviar dados ao lado servidor. Algumas bibliotecas do Javascript são utilizadas para visualização de dados, como C3.js para gráficos gerados dinamicamente e JQGrid para a tabela com dados provenientes das consultas sobre o IQA dos pontos.

O armazenamento de dados do sistema é no lado servidor, com a utilização do SGBD PostgreSQL. A estrutura do banco segue o padrão de um DW floco de neve [KIMBALL 2013], com vistas a eventualmente permitir a consulta utilizando diferentes granularidades sobre a dimensão de tempo. Seus domínios estão subdivididos em datamarts para água, clima e fauna, com diversas tabelas fato e dimensão para cada domínio. Para este trabalho a tabela fato utilizada foi a de medições de qualidade da água, com registros datando do ano 2000 até 2019. Nela estão dispostas colunas de informação temporal (data da coleta, data de inserção e data de análise), campos pertinentes aos valores, limites destes valores e campos relacionados a tabelas dimensão

1 Agradecemos as empresas Brookfield, Ceran, Certel e Hidrotérmica pelo fomento ao contínuo desenvolvimento do SIA e pelo apoio à pesquisa.

pertinentes ao ponto de coleta, qual parâmetro foi coletado, qual o método de análise e qual o de coleta, além do laboratório responsável.

O processamento destes dados e o cálculo do IQA ocorre no lado servidor a cada consulta, em vistas do número de pontos de monitoramento de água presentes no sistema e na existência de campanhas de monitoramento contínuas, sendo esta uma funcionalidade implementada após a estruturação do SIA e do DW. Após isto, os resultados são fornecidos ao gráfico gerado com a biblioteca C3.js e ocorre seu envio ao lado cliente, junto da tabela de consulta, utilizando os mesmos dados que serão dispostos no gráfico.

3. Resultados

A ferramenta desenvolvida, intitulada IQA Multipontos, é constituída inicialmente de uma tabela de seleção de pontos (Figura 1), alimentada com dados dos pontos pertinentes ao módulo de água do SIA por meio de uma consulta ao DW. O usuário pode selecionar os pontos por meio da opção de seleção automática, utilizando os parâmetros especificados em cada uma das duas caixas de seleção, escolhendo um ponto pertencente a um recurso hídrico, município, sub-bacia ou empreendimento de sua escolha, com vistas a permitir diferentes granularidades de consulta. Esta seleção passa por um processamento em Javascript, no lado cliente, e então é feita uma requisição ao lado servidor, onde uma classe controladora em PHP solicita ao DW os dados necessários e retorna-os ao lado cliente, onde alimentam uma tabela. Neste caso, ocorre na consulta a verificação no banco se as coordenadas de localização do ponto estão contidas nos pontos geográficos que formam o polígono do ponto escolhido no DW. Além disso, o usuário pode selecionar manualmente os pontos desejados.

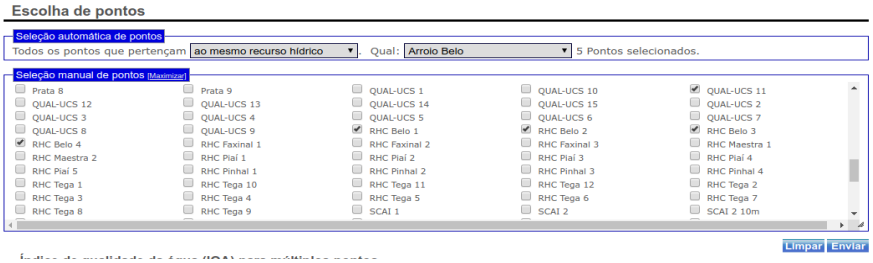


Figura 1. Tabela de seleção de pontos.

Após o envio da consulta, são gerados os componentes de visualização dos resultados (Figura 2). Para esta geração ocorre a aplicação do cálculo do IQA sobre os pontos solicitados, com dados provenientes do DW. O gráfico apresenta historicamente o resultado deste cálculo para os diferentes pontos de amostragem selecionados, ou seja, para cada ponto de amostragem é gerado uma linha no gráfico, onde cada ponto sobre ela é uma coleta com um valor de IQA calculado, em função do tempo, disposto no eixo X. Ao posicionar o mouse sobre um determinado ponto no gráfico, é possível visualizar essas informações para a coleta selecionada e para as outras realizadas na mesma data. Vale destacar que os pontos de amostragem variam quanto a quantidade e periodicidade das coletas. As diferentes faixas de cores presentes atrás do gráfico enquadram cada IQA calculado em uma classificação dos recursos hídricos de acordo com a Resolução CONAMA 357, de acordo com os valores dos parâmetros de qualidade da água. A tabela gerada, apresentada na Figura 2, é uma outra forma de exibição dos dados históricos presentes no gráfico. Nela o usuário pode visualizar o IQA calculado para cada campanha de monitoramento presente no gráfico, com o mesmo padrão de cores aplicado sobre o gráfico.

Ponto	Data Coleta	Valor	Qualidade	OD-Nota	OD-Qual
QUAL-UCS 11	31/03/2011	43.22	Ruim	50	1.94
QUAL-UCS 11	30/05/2011	52.73	Regular	96.92	2.18
QUAL-UCS 11	10/08/2011	48.15	Ruim	94.87	2.17
QUAL-UCS 11	26/10/2011	40.14	Ruim	95.04	2.17
QUAL-UCS 11	07/12/2011	28.56	Ruim	95.93	2.17
QUAL-UCS 11	01/02/2012	27.21	Ruim	82.89	2.12
RHC Belo 1	25/09/2012	52.49	Regular	86.76	2.14

Figura 2. Componentes de visualização dos resultados.

4. Considerações Finais

A funcionalidade desenvolvida permite ao usuário a visualização agrupada do IQA em múltiplos pontos. Desta forma, foi facilitada a comparação da qualidade da água em diferentes pontos da Bacia Hidrográfica. Essa vantagem diminui o tempo de trabalho do gestor ou pesquisador que busca utilizar dessas informações em suas pesquisas e estudos relacionados ao meio ambiente da região, aumentando a eficiência do SIA como ferramenta de apoio a gestão ambiental e a geração de conhecimento. Futuramente, pretende-se inserir à funcionalidade um filtro de consulta temporal, no qual o usuário poderá definir a data de início e fim do período para o qual deseja gerar o gráfico. Além disso, pretende-se acrescentar a visualização em múltiplos pontos para outros índices presentes no SIA.

Referências

ANA - Agência Nacional de Águas. Indicadores de Qualidade - Índice de Qualidade das Águas (IQA). Disponível em: <http://pnqa.ana.gov.br/indicadores-indice-aguas.asp x>. Acesso em: 08 ago. 2018.

Bikakis, N. Big Data Visualization Tools Encyclopedia of Big Data Technologies, Springer 2018. Disponível em: <https://arxiv.org/pdf/1801.08336.pdf>. Acesso em: 18 fev. 2019.

Günther, O. Environmental information systems. Acm Sigmod Record, v. 26, n. 1, p.3-4, mar. 1997.

Kimball, R. e Ross, M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. 3. ed. Editora: John Wiley & Sons, 2013. 564 p.

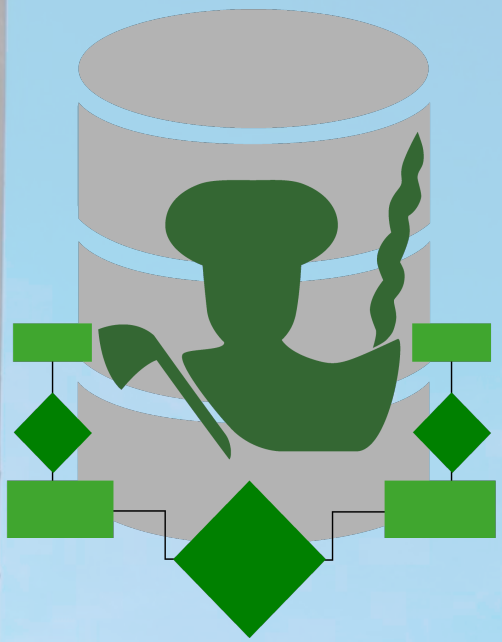
Von Sperling, M. Estudos e modelagem da qualidade da água de rios. 1 ed. v. 7. Belo Horizonte: Departamento de Engenharia Sanitária e Ambiental; Universidade Federal de Minas Gerais, 2007

O'Brien, J. A. e Marakas, G. M. Management Information Systems. Dias Technology Review, v. 4, n. 2, p.102-112, out. 2008.

Sommerville, I. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.
Elmasri, R. e Navathe, S.B. Sistemas de Bancos de Dados. 7. ed. São Paulo: Pearson Education do Brasil, 2018. 1127 p.

Cao, L. Data science. Communications Of The Acm, [s.l.], v. 60, n. 8, p.59-68, 24 jul. 2017. Association for Computing Machinery (ACM). http://dx.doi.org/10.1145/3015456.

Moore, J. Data Visualization in Support of Executive Decision Making. Interdisciplinary Journal Of Information, Knowledge, And Management, [s.l.], v. 12, p.125-138, 2017. Informing Science Institute. http://dx.doi.org/10.28945/3687.



ANAIS

XV ERBD

Escola Regional de Banco de Dados

2019

Chapecó - SC

INTELIGÊNCIA DE DADOS

Organização:

UNIVERSIDADE FEDERAL DA FRONTEIRA SUL

UNOESC

Realização:

SBC Sociedade Brasileira de Computação

Patrocínio:

CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico

CRESOL Crédito para seus sonhos

amplimed Gestão de Clínicas

Apoio:

NTIC Núcleo das Empresas de Tecnologia da Informação e Comunicação

deatec Associação Polo Tecnológico do Oeste Catarinense

www.sbc.org.br/erbd2019

/erbd.sbc