HAL
open science

Actes de la conférence BDA 2015 : Gestion de données - principes, technologies et applications

Sihem Amer-Yahia, Omar Boucelma, Elisabeth Murisasco, Noël Novelli

▶ **To cite this version:**

Sihem Amer-Yahia, Omar Boucelma, Elisabeth Murisasco, Noël Novelli (Dir.). Actes de la conférence BDA 2015 : Gestion de données - principes, technologies et applications. 2015. hal-01629448

HAL Id: hal-01629448

https://inria.hal.science/hal-01629448v1

Submitted on 6 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

BDA 2015

Gestion de Données – Principes, Technologies et Applications

29 septembre au 2 octobre 2015
Île de Porquerolles

Actes de la conférence BDA 2015
Conférence soutenue par l'Université de Toulon,
Aix Marseille Université, le CNRS, les laboratoires LSIS et LIF,
FRIIAM, MaDICs, et INRIA

Site de la conférence : `http://bda2015.univ-tln.fr/`
Actes en ligne : `https://hal.inria.fr/BDA2015`
ISSN : 2429-4586

Message des organisateurs

La conférence sur la « Gestion de Données — Principes, Technologies et Applications » (BDA 2015) est le rendez-vous incontournable de la communauté de la gestion de données en France.

La conférence a célébré son 30ème anniversaire à Autrans en 2014. La 31ème édition de la conférence a eu lieu sur l'île de Porquerolles du 29 septembre au 2 octobre 2015. Elle est co-organisée par les laboratoires CNRS LSIS (UMR CNRS 7296) et LIF (UMR CNRS 7279). La recherche en gestion de données connaît une nouvelle ère : le data-driven. Les données sont de plus en plus disponibles ce qui donne naissance à une nouvelle approche de recherche en gestion de données : comment partir de la donnée, l'organiser et la préparer pour en extraire de la valeur. De plus, beaucoup de chercheurs en gestion de données travaillent avec des industriels. BDA est une bonne occasion pour les chercheurs pour échanger autour de leurs expériences avec des jeux de données réels qui viennent de disciplines diverses et également une occasion pour les industriels pour trouver les futurs data scientists.

BDA 2015 est donc aux couleurs data-driven et industrielles. Elle est l'occasion de montrer les contributions récentes s'adressant aux nouveaux défis de recherche posés par les approches de recherche data-driven et les collaborations industrielles. Les thématiques de la conférence couvrent un large spectre de problèmes liés à la collecte, l'interrogation, la gestion et l'intégration de données dans les applications industrielles et scientifiques traditionnelles ainsi que dans les domaines tels la santé, l'environnement, la gestion des transports et réseaux urbains et électriques, et les sciences humaines et sociales.

Cette année, BDA a offert aux auteurs le choix de publier leurs articles dans les actes ou pas. On trouve donc dans ces actes que les articles que les auteurs ont souhaité publier.

Omar Boucelma (président des journées)
Sihem Amer-Yahia (présidente du comité de programme)
Elisabeth Murisasco et Noël Novelli (présidents du comité d'organisation)

Table des matières

1 Comités de BDA 2015 **4**

2 Conférenciers invités **6**
2.1 Web sémantique : beaucoup de données, quelques connaissances et un peu de raisonnement
François Goasdoué et Marie-Christine Rousset 6
2.2 Accommoder les miettes de données : Ingrédients, Recettes et Astuces
Amélie Marian et Arnaud Sahuguet 6
2.3 Data integration challenges raised by self-service Business Intelligence
Eric Simon (SAP) 7

3 Tutoriels **7**
3.1 Data cleaning in the big data era
Paolo Papotti and Jorge Quiané-Ruiz, Qatar Computing Research Institute (QCRI) 7

4 Articles longs **9**
4.1 ADVIZ.Eolas : restitution et pilotage de process temps réel data-driven
Ibrahim Safieddine, Noël De Palma and Gérald Dulac 9
4.2 Video spatio-temporal ltering based on cameras and target objects trajectories – Videosurveillance framework
Dana Codreanu, Vincent Oria, André Peninou and Florence Sèdes 15
4.3 Bi-temporal Query Optimization Techniques in Decision Insight
Azhar Ait Ouassarah, Nicolas Averseng, Xavier Fournet, Jean-Marc Petit, Romain Revol and Vasile-Marian Scuturici 20
4.4 An approach for measuring RDF data completeness
Fayçal Hamdi and Samira Si-Said Cherfi 32

5 Articles de doctorant-e-s **42**
5.1 Sara El Hassad
Interrogation par analogie dans les bases de données 42
5.2 Yifan Li
Scalable Approaches for Recommendation in Social Networks 44

6 Démonstrations **46**
6.1 Luis Galárraga
Interactive Rule Mining in Knowledge Bases 46

1 Comités de BDA 2015

Président des Journées
Omar Boucelma (LSIS), Aix Marseille Université

Présidente du comité de Programme
Sihem Amer-Yahia, Laboratoire d'Informatique de Grenoble (LIG)

Président du comité des démonstrations
Fabian Suchanek, Télécom ParisTech

Président du comité de la session doctorant-e-s
Vincent Leroy (LIG), Université de Grenoble

Présidents du comité d'organisation
Elisabeth Murisasco (LSIS), Université de Toulon
Noël Novelli (LIF), Aix Marseille Université

Comité d'organisation
Emmanuel Bruno (LSIS), Université de Toulon
Alain Casali (LIF), Aix Marseille Université
Vincent Martin (LSIS), Université de Toulon

Comité de Programme
Reza Akbarinia, INRIA Sophia Antipolis
Mohamed-Amine Baazizi, LIP6/Université Pierre et Marie Curie
Khalid Belhajjame, LAMSADE/Université Paris Dauphine
Christophe Bobineau, LIG/Institut National Polytechnique de Grenoble
Boughanem Mohand, IRIT/Université Paul Sabatier Toulouse
Pierre Bourhis, LIFL/INRIA Lille CNRS
Amel Bouzeghoub, Télécom Sud Paris/CNRS UMR 5157 SAMOVAR
Bogdan Cautis, LRI/Université de Paris-Sud
Dario Colazzo, LAMSADE/Université Paris Dauphine
Camelia Constantin, LIP6/Université Pierre et Marie Curie
Thierry Delot, LAMIH/Université de Valenciennes, INRIA-Lille
Cedric Du Mouza, CNAM
Beatrice Finance, PRiSM Laboratory/Versailles University
David Gross-Amblard, ISTIC/IRISA/Université Rennes 1
Abdelkader Hameurlain, IRIT/Université Paul Sabatier
Lotfi Lakhal, LIF/Aix Marseille Université
Philippe Lamarre, LIRIS/INSA Lyon
Anne Laurent, LIRMM/Université Montpellier 2, CNRS
Sofian Maabout, LaBRI/Université de Bordeaux
Ioana Manolescu, INRIA Saclay – Île-de-France et LRI/Université Paris Sud-11
Pascal Molli, LINA/Université de Nantes
Amedeo Napoli, LORIA
Noël Novelli, LIF/Aix Marseille Université, CNRS
Benjamin Nguyen, INSA Centre Val de Loire
Philippe Pucheral, INRIA/Université de Versailles Saint-Quentin-en-Yvelines
Daniel Rocacher, Irisa

Maguelonne Teisseire, LIRMM/Université Montpellier 2, CNRS
Farouk Toumani, LIMOS/Université Blaise Pascal , Clermont-Ferrand
Federico Ulliana, LIRMM/Université Montpellier 2, CNRS
Nicolas Anciaux, INRIA Paris-Rocquencourt

Comité des démonstrations

Alexandre Termier, Université de Rennes 1
Benjamin Nguyen, INSA-CVL
Camelia Constantin, LIP6
Danai Symeonidou, Télécom ParisTech
David Gross-Amblard, IRISA
Fabrice Jouanot, IMAG
Francesca Bugiotti, INRIA
Mauro Sozio, Télécom ParisTech
Nicoleta Preda, Université de Versailles Saint-Quentin-en-Yvelines
Pierre Bourhis, Lille1
Sarah Cohen-Boulakia, LRI
Xiao Bai, Yahoo ! Research

Vérification des plagiats

Pierre Senellart, Télécom ParisTech

Edition des actes

Fabian Suchanek, Télécom ParisTech
David Gross-Amblard, IRISA/Université de Rennes 1

2 Conférenciers invités

2.1 Web sémantique : beaucoup de données, quelques connaissances et un peu de raisonnement
François Goasdoué et Marie-Christine Rousset

Présentateurs : François Goasdoué est Professeur en Informatique à l'Université de Rennes 1. Ses travaux de recherche sont menés à l'interface des Bases de Données et de la Représentation des Connaissances et Raisonnement ; ils portent sur la gestion efficace de données (consistance, interrogation, mise-à-jour, etc) dans le cadre des graphes RDF et des bases de connaissances OWL2, et dans des architectures centralisées, décentralisées et massivement parallèles. Ses résultats sont régulièrement publiés dans les revues et conférences majeures de Bases de données et d'Intelligence Artificielle.

Marie-Christine Rousset is a Professor of Computer Science at the University of Grenoble. Her areas of research are Knowledge Representation, Information Integration, Linked Data and the Semantic Web. She has published around 100 refereed international journal articles and conference papers, and participated in several cooperative industry-university projects. She received a best paper award from AAAI in 1996, and has been nominated ECCAI fellow in 2005. She has served in many program committees of international conferences and workshops and in editorial boards of several journals.

Abstract : Le Resource Description Framework (RDF), standard du W3C pour le Web Sémantique, suscite un intérêt croissant de la part de la communauté Bases de Données. Ce modèle de données est en effet particulièrement adapté à la représentation de Big Data (données très volumineuses, hétérogènes et incomplètes) et a déjà une incarnation phare dans le Linked Data.

RDF est un modèle flexible qui permet d'exprimer de manière uniforme, sous forme de triplets, des métadonnées sur des entités référencées par des URIs, mais aussi des connaissances sur le schéma des classes et des propriétés, qui constituent ce qu'on appelle souvent des ontologies.

La mise en œuvre du Web sémantique consiste à exploiter ces connaissances par des algorithmes de raisonnement pour compléter par inférence l'ensemble des réponses à des requêtes, et aussi pour enrichir et lier les données de plusieurs sources.

Dans cet exposé, nous soulignerons les similitudes et les spécificités du modèle RDF par rapport aux modèles formels des bases de données déductives et des bases de données incomplètes. Nous dégagerons ensuite les défis découlant de ces spécificités pour mettre en œuvre des techniques efficaces d'interrogation ainsi que de liage de données et de connaissances. Nous présenterons enfin les principales approches proposées dans la littérature récente pour relever certains de ces défis.

2.2 Accommoder les miettes de données : Ingrédients, Recettes et Astuces
Amélie Marian et Arnaud Sahuguet

Présentateurs : Amélie Marian is an Associate Professor in the Computer Science Department at Rutgers University. Her research interests are in Personal Information Management, Ranked Query Processing, Semi-structured data and Web data Management. Amélie received her Ph.D. in Computer Science from Columbia University in 2005. From March 1999 to August 2000, Amélie was a member of the VERSO project at INRIA-Rocquencourt. She received B.S. and M.S. degrees from Université Paris Dauphine, France in 1998 and 1999, respectively. She is the recipient of a Microsoft Live Labs Award (2006), three Google Research Awards (2008, 2010, and 2012) and an NSF CAREER award (2009).

Dr Arnaud Sahuguet is a technologist and entrepreneur with a passion to invent, architect and build products that leverage technology to solve meaningful problems and have a large social impact. His goal is to empower people and organizations to be more productive and collaborative through innovation. Before joining GovLab as Chief Technology Officer, Arnaud spent 8 years at Google as a product manager for speech recognition and Google Maps ; he founded and launched the OneToday mobile fundraising platform for Google.org ; he also worked on child protection and civic innovation. Before Google, he spent 5 years at Bell Labs research as member of technical staff working on standardization, identity management and

converged services. Arnaud holds a PhD in Computer Science from Univ. of Pennsylvania, a MSc from Ecole Nationale des Ponts et Chaussées and a BSc from Ecole Polytechnique in France.

Abstract : Big Data has become a hot topic in the last few years in both industry and the research community. For the most part, these developments were initially triggered by the requirements of Web 2.0 companies. Both technical and non-technical issues have continued to fuel the rapid pace of developments in the Big Data space. Open source and non-traditional software entities have played key roles in the latter. As it always happens with any emerging technology, there is a fair amount of hype that accompanies the work being done in the name of Big Data. The set of clear-cut distinctions that were made initially between Big Data systems and traditional database management systems are being blurred as the needs of the broader set of ("real world") users and developers have come into sharper focus in the last couple of years. In this talk, I will survey the developments in Big Data and try to distill reality from the hype !

2.3 Data integration challenges raised by self-service Business Intelligence *Eric Simon (SAP)*

Abstract : Enterprise Business Intelligence (BI) traditionally provides solutions to business users for managed reporting (ad-hoc query and reporting or pixel-perfect reporting), dashboards and data analysis. BI solutions heavily rely on the IT organization to create the data warehouse and data marts underpinning the BI system, as well as the semantic layers specifically designed over this trusted data foundation to model information used by reports, dashboards and analytic queries. A decade ago, BI has evolved to empower business users to create personalized reports and analytical queries, and let them manipulate and explore information directly, without resorting to IT. Business users and analysts are now demanding access to true "self-service" capabilities beyond data discovery and rich interactive visualization of IT-curated data sources, to include access to sophisticated data integration tools to prepare their data for analysis, and data governance capabilities. This growing demand raises the need for new data-driven and iterative solutions better suited to business users than the traditional « design-test-deploy » paradigm typically adopted by IT organizations. In this paradigm shift, business users « model their data as they go » creating their own analyses, reports and performance indicators. Business users need new powerful data-driven and interactive user interfaces as well as new capabilities to search for data, easily assess the quality of data, semi-automate the curation, profiling, and enrichment of data, and suggest how to expand and combine datasets that are semantically related depending on the user interaction context and profile. This talk will review the requirements of « self-service BI » and explain the technical challenges it raises to provide more data-driven data integration solutions. Some of the recent directions taken by SAP in this field will be outlined and illustrated. Open issues will be presented at the end.

3 Tutoriels

3.1 Data cleaning in the big data era *Paolo Papotti and Jorge Quiané-Ruiz, Qatar Computing Research Institute (QCRI)*

Abstract : In the « big data » era, data is often dirty in nature because of several reasons, such as typos, missing values, and duplicates. The intrinsic problem with dirty data is that it can lead to poor results in analytic tasks. For instance, Experian QAS Inc. reported that poor customer data cost British businesses £8 billion loss of revenue in 2011. Therefore, data cleaning is an unavoidable task to have reliable data for final applications, such as querying and mining. Data cleaning (a.k.a. data preparation) is a popular activity in both industry and in academia. Nevertheless, data cleaning is hard in practice as it requires a great amount of manual work. Several systems have been proposed to achieve the level of automation and scalability required by the volume and variety in big data. They rely on a formal, declarative approach based on first order logic : users provide high-level specifications of their tasks (the « what ») ; the systems compute optimal solutions without human intervention on the generated code (the « how »). However, despite the positive

results in automating the data cleaning task, the volume (scalability) and variety of big data remain two open problems. In this tutorial, we first describe recent results in tackling data cleaning with a declarative approach. We then discuss how this experience has pushed several groups to explore a new approach to the problem to deal with the volume and variety of big data. In particular, we discuss how user defined functions and declarative specifications can coexist in a unified system, ultimately taking the best from both worlds.

ISSN 2429-4586

ADVIZ.Eolas : Restitution et Pilotage de Process Temps Réel Data-Driven

Ibrahim Safieddine*
Business & Decision Eolas
Université Grenoble Alpes
Grenoble, France
ibrahim.safieddine@businessdecision.com

Noël De Palma†
Université Grenoble Alpes
220 Rue de la chimie
Grenoble - France
noel.depalma@imag.fr

Gérald Dulac‡
Business & Decision Eolas
29 Rue Servan
Grenoble, France
gerald.dulac@business-decision.com

ABSTRACT

Nous présentons un outil d'analyse, de diagnostic, de calcul et de visualisation des données provenant de milliers de capteurs hétérogènes, utilisable par des process quelconques et testé sur un Data Center en production. Grâce à un historique complet et des mesures temps réel, cet outil offre une vue globale de l'état de l'ensemble des infrastructures FaaS - IaaS - PaaS et SaaS et des process ITIL (Information Technology Infrastructure Library), pour l'analyse complexe et la détection des comportements anormaux, l'anticipation des opérations de maintenance et la baisse du taux de panne en maintenant en condition optimale les équipements.
Les travaux présentés décrivent l'architecture modulaire de l'outil appelé ADVIZ.Eolas. Le module d'acquisition des mesures hétérogènes à des fréquences variables en temps réel, le module de stockage Big Data des données brutes et calculées et le module de visualisation utilisant les technologies Web, représentent les modules de base de l'outil qui se connecte aussi aux standards du marché ITIL (CMDB (Configuration Management Database) et LDAP (Lightweight Directory Access Protocol)). Des Modules de calcul Big Data temps réel et différé se connectent au module de stockage pour la génération d'indicateurs de défaillance et la détection des anomalies. Cet outil est issu de l'expérience de divers projets de recherche passés et en cours.

General Terms

Big Data, Data-Driven

*Doctorant Cifre chez Eolas, membre de l'équipe ERODS du LIG (Laboratoire informatique de Grenoble)

†Professeur à l'université Grenoble Alpes. Directeur de l'équipe ERODS du LIG (Laboratoire informatique de Grenoble)

‡Fondateur d'Eolas, directeur du programme de recherche Datalyse pour le compte d'Eolas, représentant d'Eolas pour les programmes EnergeTIC et CtrlGreen

(c) 2015, Copyright is with the authors. Published in the Proceedings of the BDA 2015 Conference (September 29-October 2, 2015, Ile de Porquerolles, France). Distribution of this paper is permitted under the terms of the Creative Commons license CC-by-nc-nd 4.0.
(c) 2015, Droits restant aux auteurs. Publié dans les actes de la conférence BDA 2015 (29 Septembre-02 Octobre 2015, Ile de Porquerolles, France). Redistribution de cet article autorisée selon les termes de la licence Creative Commons CC-by-nc-nd 4.0.
BDA 29 septembre au 2 octobre 2015 - Île de Porquerolles - France.

Keywords

Stockage Distribué, Data Center, Green Computing, Visualisation de données, Efficience Énergétique

1. INTRODUCTION

De plus en plus d'hébergeurs reconnaissent la valeur ajoutée des applications de gestion de l'infrastructure des centres de données (DCIM [1]). Le besoin en DCIM est généré par la réalité des centres de données modernes : une automatisation grandissante, une augmentation importante en terme de volume de données stockées et de puissance de calcul et une demande exigeante en continuité de meilleur niveau de qualité de service SLA (Service Level Agreement) [2]. De ce fait, la mise en place de fonctionnalités de monitoring et de reporting est le premier pas vers une meilleure compréhension et optimisation du fonctionnement global. Ces outils sont utilisés pour surveiller l'état des équipements, détecter différents problèmes, lever des alertes, prévenir le personnel en charge et fournir un rendu visuel temps réel et historisé afin d'effectuer des analyses et des prévisions stratégiques et opérationnelles. Ces outils doivent faire face aux contraintes que la réalité physique de l'infrastructure implique : l'infrastructure d'un Data Center n'est pas figée, bien au contraire, la plupart des Data Center voient leur capacité augmenter au fil des ans et il est donc indispensable d'avoir des outils capables de faire face automatiquement à ces modifications. Enfin la conservation des données relatives au fonctionnement d'un Data Center représente un grand volume qu'il faut prendre en compte technologiquement, au risque, à terme, de devoir agréger les archives et de perdre en finesse d'analyses. Business & Decision Eolas est un hébergeur qui gère trois Data Center sur Grenoble, dont un Data Center Green ayant une capacité de 13.000 serveurs. Plus de 5000 points de mesures sont déployés sur le Data Center, ce qui génère plus de 3 Millions de mesures par jour. Les mesures des différents capteurs sont centralisées mais seulement une petite partie est exploitée directement au travers d'alertes, de tableaux de contrôle ou bien des différents indicateurs.

Le but de ce projet est donc de développer une application de gestion d'infrastructure hétérogène et dynamique intégrant des technologies Big Data afin de conserver une grande précision sur les données sans nuire aux performances de l'outil. Elle permettra notamment, sur l'environnement Eolas, l'analyse et la visualisation des données, en vue de réaliser des diagnostics, la surveillance générale et l'optimisation

1. Data Center Infrastructure Management

du bon fonctionnement de l'infrastructure dans son intégralité. La surveillance et le contrôle offerts par un tel outil permettent de pérenniser l'exploitation de l'infrastructure, d'optimiser le travail des équipes de gestion et de fournir un socle solide pour la prise de décision vis à vis de l'ensemble de l'environnement. L'architecture de l'application doit pouvoir s'adapter à un panorama d'outils et d'environnements variés (bâtiments d'habitation, Open-data...).

2. CONTEXTE ET MOTIVATION

2.1 Contexte

L'émergence des technologies Big Data permet actuellement la mise en place de gros entrepôts de données facilement mais l'analyse de ces données de manière intelligente et pertinente, afin de fournir une information autre que celle fournis par les données brutes, se fait difficilement. Eolas ainsi que plusieurs laboratoires se sont associés, dans un projet nommé Datalyse[4], afin d'y faire face. Ce projet vise à mettre en place des traitements intelligents de Big Data pour la collecte, la certification, l'intégration, la catégorisation, la sécurisation, l'enrichissement et le partage de données hétérogènes Big Data.

Avec le projet CtrlGreen [3] Eolas et ses partenaires chercheurs ont mis en place des boucles de contrôle qui permettent d'optimiser la consommation électrique, surtout au niveau du refroidissement, en fonction de divers paramètres collectés au sein de l'environnement des Data Center. Il est donc primordial de surveiller ces nouvelles boucles et de remonter de nouvelles mesures et indicateurs pour valider leur bon fonctionnement.

Actuellement Eolas utilise l'outil ScopeBR, qui permet de visualiser les différents mesures et indicateurs liés à l'exploitation de ses Data Center. L'architecture de ScopeBR permet de prendre en compte les données issues de tous types d'équipements : Compteurs, Disjoncteurs, Sondes, Capteurs (contacts secs, modbus...), Bases de données externes, et ainsi de restituer les informations sous forme de tableaux de bords standards. Les données collectées par ScopeBR sont stockées dans une base de données relationnelle, agrégées au fur et à mesure pour éviter de saturer la base. Cette baisse significative de la taille des données induit une baisse de la précision des anciennes données, car l'agrégation se fait sur des données anciennes pour conserver une granularité fine sur les données récentes, potentiellement plus importantes.

La difficulté est de trouver les outils et les techniques permettant d'optimiser les temps de traitement afin de pouvoir se rapprocher le plus possible du temps réel tout en gardant une granularité fine des mesures.

2.2 Motivation

L'efficience énergétique des Data Center, ayant une architecture très complexe, est devenu une préoccupation importante pour tous les gestionnaires de Data Center. Un Data Center est composé de nombreux équipements hétérogènes et nécessite un outil de surveillance pour augmenter sa fiabilité et son rendement, et donc baiser les coûts d'exploitation. Un outil centralisé permet de réaliser des diagnostics sur des pannes ou des dysfonctionnements pour prédire l'impact de modification d'un paramètre ou d'un équipement sur les autres équipements du Data Center. L'outil devra donc rendre accessible l'ensemble des équipements qui composent le Data Center, leurs mesures et leurs états afin d'aider à la prise de décision.

Les travaux de recherche menés par Business & Decision Eolas, l'ont permis de devenir un des premiers labellisés "Code of Conduct" de l'Union Européenne et à mettre en place la mesure des indicateurs de fonctionnement des Data Center de "GreenGrid" dans le cadre du projet de recherche EnergeTIC [5] et à en établir un nouveau dans le cas précis des Data Center d'Eolas : PUE (Power Usage Effectiveness), WUE (Water Usage Effectiveness), CUE (Carbon Usage Effectiveness), ERE (Energy Reuse Effectiveness) et CPE (Computer Power Efficiency). Le nouvel indicateur EUE_cpu se base sur le modèle d'efficience du PUE. Il permet de mesurer l'efficacité d'usage des ressources informatiques en activité. Eolas a par ailleurs décidé de rendre public en temps réel ces indicateurs pour son propre Data Center Mangin.

Pour Eolas, il est important de disposer d'un système d'analyse, de diagnostic et de reporting du Data Center Mangin qui remplace le logiciel ScopeBR, et qui sera généralisé dans un système couvrant toute l'activité de l'hébergement. Ce logiciel permet de collecter des données, les stocker et les restituer en utilisant les technologies Big Data sur des milliers de points de mesure, équivalent à plus de 3 Millions de mesures collectées par jour sur les équipements DCM tout en gardant une granularité très fine des mesures et des indicateurs. Ce nouveau logiciel sera ensuite utilisé comme base de logiciel de pilotages de process. Ce nouveau logiciel se base sur des solutions 100% Open Source avec la possibilité de stockage de très gros volumes de données (plusieurs To, Po) et un accès rapide sur de longs historiques (plusieurs années).

3. APPROCHE ET CONTRIBUTION

Nous présentons une architecture mettant en oeuvre un système d'acquisition de données en temps réel provenant d'équipements très hétérogènes (FaaS - IaaS - PaaS - SaaS) couplé à un système de stockage Big Data pour accomplir quatre objectifs principaux :

- Collecter : les données de l'ensemble des composants qui constituent un Data Center (e.g., Capteurs de température et d'humidité).
- Maîtriser : à travers ses propres indicateurs ou utiliser les standards (PUE, CUE, WUE...), avec la possibilité de calcul rétroactif sur les historiques.
- Surveiller : ses propres indicateurs pour maintenir en condition optimale les équipements et anticiper les opérations de maintenance.
- Historiser : les métriques qui permettent de valoriser les écarts et détecter les comportements anormaux en conservant une granularité fine des mesures et des indicateurs (données brutes).

Dans ce document, nous présentons l'outil ADVIZ pour **A**nalyze, **D**iagnostic et **VIZ**ualisation :

- Analyse : Analyse temps réel des comportements anormaux, analyse prédictive de panne, reporting des incidents, anticipation des opérations de maintenance et maintien en condition optimale des équipements.
- Diagnostic : Traçabilité, historiques illimité des incidents, corrélation entre différents indicateurs et impact de la variation d'un indicateur sur l'ensemble du DC (dérives lentes, cycles...).
- Visualisation : Suivi quotidien des équipements des différentes équipes et suivi de nos propres indicateurs

ISSN 2429-4586

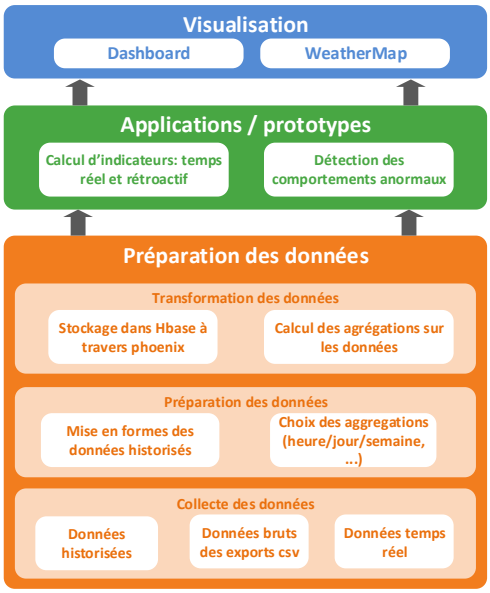

Figure 1: ADVIZ.eolas - Modélisation

calculés ou des standards (PUE, WUE, EUE...), sous formes de tableaux de bord ou des WeatherMaps temps réel.

Le travail effectué est donc en vue de mettre en place une architecture modulaire qui permet d'effectuer des traitements Big Data sur les données existants d'un Data Center, et fournir aux intervenants une visualisation temps réel des mesures et des indicateurs (< 1s) pour détecter au plus tôt les anomalies. Nous mettons en oeuvre des nouvelles solutions pour collecter les données, les stocker et les restituer en utilisant des bases de données NoSQL orientées Big Data.

4. ADVIZ.EOLAS

Dans cette section, nous présentons la modélisation conceptuelle de traitement des données du logiciel ADVIZ. Nous détaillerons ensuite l'architecture logicielle de l'application ADVIZ.eolas ainsi que les flux de communication entre les différents modules de l'architecture.

4.1 Modélisation

La modélisation ADVIZ est constituée de trois couches.

La couche de "Préparation des données" : Les données collectés en temps réel proviennent de sources très hétérogènes et à des fréquences très variables (e.g., 30s, 1min, 5min...). Par exemple, un Data Center est composé à la fois de systèmes industriels (automates, gestion climatique, gestion de pompes hydrauliques, gestion électrique, gestion technique du bâtiment, caméras de surveillance, panneaux solaires, etc.) et de systèmes informatiques (e.g., Serveurs, routeurs, commutateurs...). Pour garder un historique complet, une autre partie des données proviennent des historiques des mesures collectées avant la mise en place d'ADVIZ et stockés dans d'autres bases de données, ou sous forme de fichiers plats ou CSV. Ces données seront enrichis avec des mesures provenant de sources externes, e.g., température extérieure, prévisions météo.

Contrairement aux données dites traditionnelles, tel que les bases de données relationnelles, les données Big Data nécessitent plusieurs pré-traitements de nettoyage et de transformation afin de les rendre exploitables. Pour uniformiser ces données, la phase de "préparation des données" consiste à mettre les mesures historisés et temps réel sous un format structuré et compréhensible par ADVIZ et choisir les mises en formes à appliquer. Le choix des périodes et des types d'agrégations (e.g., minimum, maximum, moyenne) des mesures à mettre en place, permet un accès plus rapide aux informations utiles.

Finalement, dans la phase "Transformation des données", nous effectuons des opérations simples sur les données avant de les stocker dans la base HBase [6]. Il est important de préciser que nous gardons une copie des données bruts dans la base de données HBase. Cela permet d'accéder aux mesures avec une granularité très fine sur des historiques très éloignées.

Les "Applications/prototypes" : Dans cette couche, nous connectons les applications Big Data à l'entrepôt de données pour générer de nouvelles indicateurs ou des détecter des anomalies. La première application mise en place permet de calculer des indicateurs complexes à partir de mesures brutes ou agrégées. Un indicateur est présenté sous forme d'une formule mathématique avec des opérateurs et des mesures, on peut également utiliser d'autres indicateurs. Les applications peuvent être temps réel (e.g., Apache Storm) ou temps différé (e.g., Apache Hadoop, Apache Spark). Ces applications génèrent de nouvelles valeurs ou détectent des comportements anormaux, qui seront stockés dans des tables HBase et visibles dans la couche Visualisation.

La couche de Visualisation : Cette couche permet de représenter les mesures et les indicateurs sous forme de tableaux de bord temps réel personnalisables, ou de WeatherMap. Les tableaux de bord (cf. Figure 4) donnent une visibilité sur un ensemble des mesures et des indicateurs sous forme de graphes et de mesures temps réel. Les WeatherMaps (cf. Figure 2) donnent une vue globale de l'état de l'ensemble des équipements et des mesures. Le module de visualisation permet de choisir la période d'échantillonnage des mesures et donne la possibilité de navigation dans l'historique des mesures et des indicateurs sur de très longues périodes.

4.2 Architecture

Les besoins définis précédemment, nous ont amené à définir une architecture logicielle distribuée et scalable, capable d'analyser rapidement et efficacement des données existantes et au fil de l'eau. La figure 3 montre les différents modules de l'architecture du système ADVIZ.eolas : Un module de visualisation et de diagnostic, un module d'acquisition des données, un module de stockage Big Data NoSQL et un module de calcul qui se branche directement sur l'entrepôt des données historisées. Nous détaillons le schéma de l'architecture dans la figure 5.

L'inventaire matériel d'ADVIZ est alimenté par la CMDB, qui représente la source complète d'inventaire de tous les équipements d'un processus ITIL. Dans l'exemple d'un Data Center, la CMDB couvre tout l'inventaire allant des équipements de distribution électrique et de refroidissement en passant par le réseau et le stockage, jusqu'aux applications et services. La CMDB est maintenu à jour grâce à des logiciels de synchronisation automatique pour une vision temps réel

ISSN 2429-4586

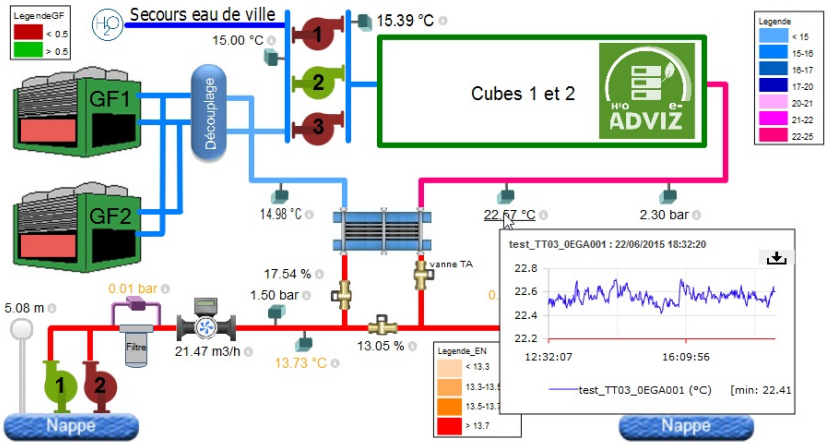

Figure 2: ADVIZ.eolas - WeatherMap de production de froid dans le Data Center de Mangin

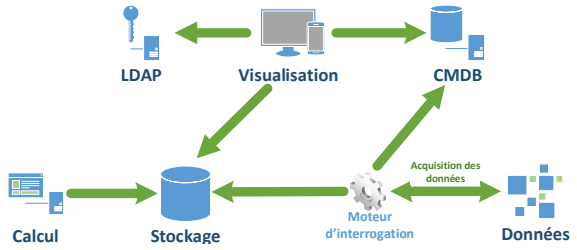

Figure 3: ADVIZ.eolas - Archicture globale

des équipements du Data Center. L'authentification et la gestion des utilisateurs passent à travers un serveur LDAP.

4.2.1 *Acquisition et stockage des données*

Le moteur d'interrogation et d'acquisition des données permet d'interroger l'ensemble des équipements, avec des protocoles variés à des fréquences différentes pour récupérer les mesures. Ce moteur est basé sur un système d'ordonnancement très complexe et sur des queues de messages implémentant le protocole MQTT (Message Queuing Telemetry Transport) pour la scalabilité et la répartition de la charge entre les différentes machines d'interrogation (Pollers). L'ordonnanceur se base sur l'inventaire de l'ensemble des points de mesures, la fréquence et la méthode d'interrogation définis dans la CMDB. Dans un Data Center, les équipements sont très hétérogènes, et donc peuvent être interrogés à travers des protocoles différentes : Mob-bus pour les équipements électriques (e.g., disjoncteur, contacteur), SNMP [2] pour les PDU (Power distribution unit) ou en HTTP [3] pour les Web Services... Ainsi il a été nécessaire d'être capable d'interroger des équipements aussi divers que des climatiseurs, des systèmes de répartition électrique, des électrovannes ou des pompes et ceci en les faisant communiquer avec des systèmes informatiques plus classiques qui se sont ensuite chargés de construire l'image globale du Data Center. La collecte des mesures peut être faite en mode PULL ou PUSH.

Les données provenant du moteur d'acquisition alimentent le module de stockage à travers des queues de messages. Le module de stockage est basé sur une base de données HBase. HBase est un système de gestion de base de données, non relationnel distribué et orienté colonnes, capable de gérer des tables avec plusieurs milliards de lignes. C'est une base de données facilement scalable. Nous avons choisis d'installer cette base de données sur le système de fichier HDFS [4] d'Hadoop pour la distribution et la réplication des données. Nous utilisons également la couche Apache Phoenix[1], qui fournit une interface SQL au-dessus de HBase permettant d'exécuter des requêtes SQL à faible latence sur HBase. Apache Phoenix permet d'optimiser les performances des requêtes en transformant la requête SQL en plusieurs scans HBase qui seront lancés en parallèle. Les données sont stockées sous forme de séries temporelles : <*id, valeur, timestamp*>. Malgré la faible latence des requêtes Phoenix (<100ms par mesure), l'affichage instantané de l'état et la valeur des différents équipements d'une WeatherMap est long. Pour résoudre ce problème, nous avons mis en place une base de données MySQL qui sert de cache de la dernière valeur de chaque mesure. Avec un temps de réponse inférieur à 5ms pour récupérer une mesure, l'affichage instantané des états et valeur d'une WeatherMap est instantané.

Le module de calcul Big Data temps réel et différé est connecté directement aux données brutes et calculées de la base de données HBase, pour effectuer des calculs simples ou complexes, et générer de nouvelles données et indicateurs. On peut avoir des systèmes de calcul de type MapReduce, Yarn, Storm, Spark ou autre. Nous représentons dans la suite des exemples d'applications Big Data déjà mis en place dans le module de calcul.

4.2.2 *Visualisation des données*

Le module de visualisation récupère les données brutes et calculées, depuis le module de stockage, à travers un web service REST (Representational State Transfer). Ces données sont ensuite représentées sous forme graphiques, des WeatherMaps, de tableaux de bord...

2. Simple Network Management Protocol
3. Hypertext Transfer Protocol
4. Hadoop Distributed File System

ISSN 2429-4586

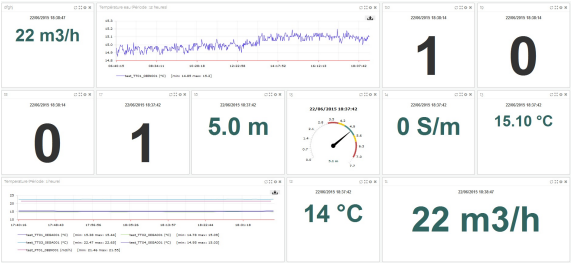

Figure 4: ADVIZ.eolas - Tableau de bord

La figure 2 montre la WeatherMap du système de refroidissement du Data Center de Mangin. La couleur et la photo attribuées à chaque module changent en fonction de sa valeur en temps réel, pour avertir l'utilisateur visuellement. En passant la souris sur les mesures, nous pouvons visualiser le graphe de variation de la mesure aggrégée ou brute, sur une période définis par l'utilisateur. La détection des anomalies de dysfonctionnement est donc plus simple et fiable.

La figure 4 montre le tableau de bord des mesures du système de refroidissement du Data Center d'Eolas. Les mesures sont présentées sous formes de graphes (lignes, histogrammes...), valeurs ou jauges. La couleur des mesures change en fonction de la valeur : quand une mesure dépasse le seuil d'alerte, la valeur passe au rouge. Les graphes permettent de faire des diagnostics plus précis grâce la vue globale de plusieurs indicateurs et mesures (e.g., une augmentation rapide de la température du Data Center accompagnée par une baisse du débit de l'eau).

Les mesures et les indicateurs sont collectés à des fréquences différentes, nous avons donc mis en place un système qui met à jour chaque mesure sur les tableaux de bords et les WeatherMap, en fonction de sa fréquence et de sa date dŠinterrogation de l'équipement.

4.3 Applications Big Data

Nous avons implémenté deux applications Big Data dans le module de calcul.

4.3.1 Calcul d'agrégations

Afin d'optimiser l'affichage des mesures et des indicateurs sur de longues périodes, nous avons intégré un système d'agrégations des données. Sur des graphiques représentant des périodes trop longues (e.g., 10ans), les courbes deviennent illisibles avec une granularité très fine, d'où l'intérêt d'afficher des données agrégés qui montrent l'évolution générale des valeurs et donc uniquement l'information utile. Pour avoir un retour temps réel des valeurs agrégés, les agrégations sont calculées régulièrement. L'agrégation est réalisée sur différentes périodes (e.g., 15 minutes, 1 heure, 1 journée). Logiquement, la période d'agrégation ne peut pas être inférieure à la fréquence de calcul de l'indicateur, il est donc possible que certaines périodes d'agrégation ne soient pas disponibles suivant le choix de la fréquence de calcul de l'indicateur. Pour réaliser les agrégations, il est possible d'utiliser plusieurs types d'opérations (Moyenne, Maximum, Minimum, première valeur, dernière valeur). Chaque type d'agrégation correspond à un besoin spécifique en fonction de la grandeur mesurée et du phénomène mis en avant par l'indicateur. Par exemple, pour un indicateur qui reflète un compteur, l'agrégation utilisée sera le maximum car pour un compteur la valeur qui importe est la plus élevée. Si au contraire l'indicateur reflète une puissance consommée, l'agrégation utilisée sera la moyenne sauf si l'utilisateur souhaite faire ressortir les pics de consommation et auquel cas il utilisera l'agrégation maximum.

Table 1: Cluster Hadoop de Benchmark

Nom serveur	CPU	Mémoire
Master	2 * 2 coeurs * 1.6GHz	5Go
Workers 1-3	2 * 2 coeurs * 1.6GHz	6Go

4.3.2 Calcul d'indicateurs

La création dynamique d'indicateurs est la deuxième application intégrée dans le système ADVIZ. Un indicateur est définit par une formule de calcul, intégrant des mesures provenant des données collectées et éventuellement d'autres indicateurs, avec une possibilité ouverte de création de nouveaux indicateurs en utilisant des opérateurs : +, - , *, /, AND, OR, Somme... Il est possible de lancer des calculs rétroactives sur des indicateurs à partir d'une date ultérieure (e.g., créer un indicateur et lancer le calcule sur les deux dernières années). Le calcul rétroactif utilisera l'historique des données du module de stockage, à condition que nous disposions de suffisamment d'historique des mesures pour accomplir le calcul. Les données générées seront stockés dans le module de stockage.

Exemple d'un indicateur simple :

$$indicateur_total = \sum_{i=0}^{n} Mesures_i \tag{1}$$

Exemple d'un indicateur lié à un autre indicateur :

$$indicateur_moyenne = Indicateur_total/n \tag{2}$$

4.4 Mesures de Performance

Pour vérifier la capacité du système de stockage à fournir un accès temps réel aux données historisées (< 500ms), nous avons réalisé des tests de performance pour mesurer le temps de réponse en utilisant un cluster de test Hadoop Cloudera CDH 4.5, avec HBase 0.96 et Phoenix 3.3. Le cluster de test est constitué de 4 serveurs : un serveur master hébergeant le NameNode et le HBase Master et trois workers de calcul hébergeant les DataNodes et les RegionServer de HBase. La table 1 détaille les caractéristiques physiques des serveurs du cluster.

Nous avons réalisé des comparaisons de temps réponse entre un cluster HBase, à un autre cluster HBase doté de la couche Phoenix, sur des historiques de données de capteurs provenant du Data Center de Mangin. Le temps de réponse étant dépendant de la nature des données et de leur structure. Nous avons créé une table de mesures contenant plus de 160 Millions de lignes. La table 2 montre les temps de réponse des requêtes effectuant des opération sur 6000 mesures, sur une base de données HBase et montre le gain en utilisant la couche Phoenix au dessus de HBase. Les mesures ont été effectuées avec la configuration HBase par défaut proposé par Cloudera.

L'utilisation de la couche Phoenix offre un accès, avec une très faible latence (<200ms), aux données historisés dans la base HBase. Le temps d'affichage d'une mesure, qui corres-

ISSN 2429-4586

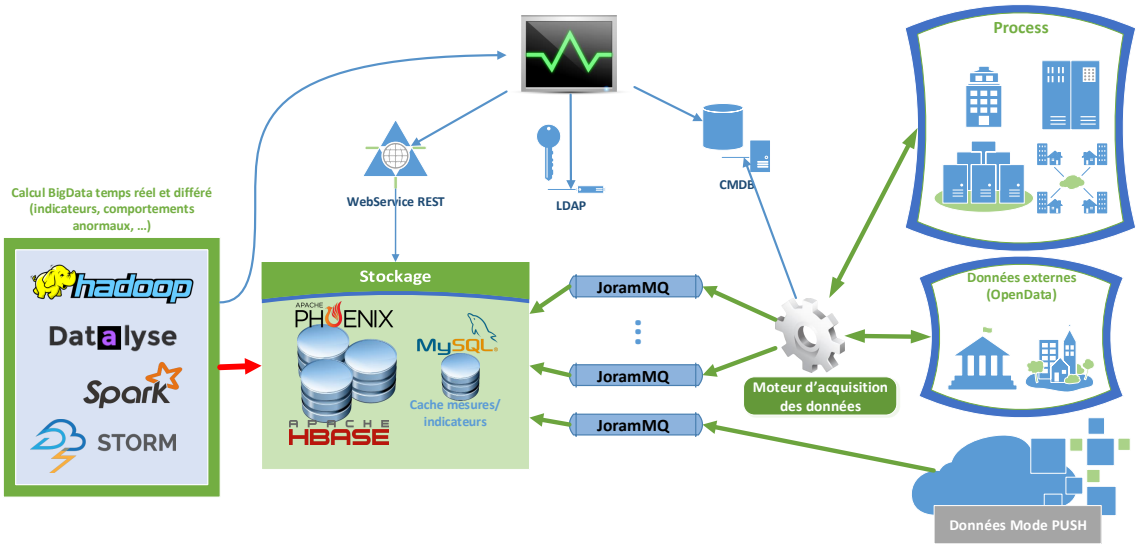

Figure 5: ADVIZ.eolas - Archicture détaillée

Table 2: Temps de réponse des requêtes

Opération	HBase	HBase avec Phoenix
Select	> 5s	0.62s
Count	> 5s	0.09s
SUM	> 5s	0.17s
AVG	> 5s	0.14s
MAX	> 5s	0.15s

pond au temps de lecture des données de la base HBase et du temps nécessaire à la librairie graphique pour générer le graphe, se fait en temps réel (<500ms). Les graphes dans un tableau de bord sont chargés en parallèle et indépendamment les uns des autres pour avoir un rendu temps réel.

5. CONCLUSIONS

Nous avons présenté une architecture distribuée et scalable, pour le stockage, la visualisation et l'analyse de très gros volumes de données. Cette architecture constitue une base de logiciels de pilotage de process. ADVIZ est actuellement utilisé chez Eolas en exploitation pour l'analyse des données provenant du Data Center de Mangin. ADVIZ est en cours d'utilisation dans d'autres projets.

Dans les perspectives du projet ADVIZ, nous travaillerons sur l'intégration de nouvelles applications dans le module de calcul pour la détection de comportements anormaux qui pourront être définis selon des patterns de seuils reliés conditionnels, exprimés avec une grammaire spéciale. Il doit être possible de spécifier des mesures ou des indicateurs, des conditions et combiner plusieurs expressions avec des opérateurs logiques. Nous envisageons en plus d'utiliser les algorithmes de jointures temporelles et hiérarchiques pour détecter de nouvelles anomalies de fonctionnement plus complexe. En plus du pattern, il doit être possible de renseigner une action à réaliser pour corriger un comportement anormal. Une fois le comportement créé, il est possible de le rattacher à un ou plusieurs équipements et également de l'ajouter dans un tableau de bord pour afficher l'historique des événements.

Deux applications directes seront mises en place avec ces nouvelles algorithmes :

— Efficacité énergétique : identification des anomalies de fonctionnement du Data Center par la collecte, le traitement et l'analyse des données des capteurs de refroidissement et de distribution électrique : e.g., une montée rapide de la température d'une salle, ou une sonde de température qui ne change pas de valeurs pendant plusieurs jours.

— Détection des comportements anormaux réseau du Data Center : e.g., une attaque réseau de type DDoS qui sature le réseau.

D'autres types d'applications, réalisés dans le cadre d'autres projets de recherche seront intégrés dans le module de calcul.

6. REMERCIEMENTS

Ces travaux de recherche sont partiellement financés par les projet de recherche CtrlGreen (ANR-11-INFR 012-11)[3] et Datalyse [4].

7. REFERENCES

[1] Apache phoenix. http ://phoenix.apache.org/. [Online ; accessed April-2015].

[2] Cloud service level agreement standardisation guidlines. Brussels, 24/06/2014.

[3] Ctrlgreen. http ://www.ctrlgreen.org/. [Online ; accessed April-2015].

[4] Datalyse. http ://www.datalyse.fr/. [Online ; accessed April-2015].

[5] Efficience des datacenters, les retombées du projet energetic. Grenoble, 21/03/2013.

[6] L. George. *Hbase : The Definitive Guide*. O'REILLY Media, United States of America, September 2011.

Video spatio-temporal filtering based on cameras and target objects trajectories - Videosurveillance framework

Dana Codreanu
IRIT, University Paul Sabatier
Toulouse, France
dana.codreanu@irit.fr

Vincent Oria
New Jersey Institute of Technology, NJ, USA
vincent.Oria@njit.edu

André Peninou
IRIT, University Paul Sabatier
Toulouse, France
andre.peninou@irit.fr

Florence Sèdes
IRIT, University Paul Sabatier
Toulouse, France
florence.sedes@irit.fr

ABSTRACT

This paper presents an ongoing work that aims at assisting videoprotection agents in the search for particular video scenes of interest in transit network. The video-protection agent inputs a query in the form of date, time, location and a visual description of the scene. The query processing starts by selecting a set of cameras likely to have filmed the scene followed by an analysis of the video content obtained from these cameras. The main contribution of this paper is the innovative framework that is composed of: (1) a spatio-temporal filtering method based on a spatio-temporal modeling of the transit network and associated cameras, and (2) a content-based retrieval based method on visual features. The presented filtering framework is to be tested on real data acquired within a French National project in partnership with the French Interior Ministry and the French National Police. The project aims at setting up public demonstrators that will be used by researchers and commercials from the video-protection community.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

Keywords

video-protection framework, spatio-temporal filtering

1. INTRODUCTION

Public and private locations nowadays rely heavily on cameras for surveillance and the number of surveillance cameras in service in public and private areas is increasing. But when needed, the content the surveillance videos is analyzed by human agents that have to spend time watching the videos organized in a matrix called video wall. Several studies have showed the cognitive overload coupled with boredom and fatigue that often lead to errors in addition of the excessive processing time. In that context, the main question is which tools can assist the human agents better do their work?

Many efforts to develop "intelligent" video-surveillance systems have been witnessed in the past years. The majority of these efforts focused on developing accurate content analysis tools [3] but the exhaustive execution of content analysis is resource intensive and gives poor results in addition because of the heterogeneity of the video content. The main idea we put forward in this paper is to use the metadata from different sources (e.g., sensor generated data, technical characteristics) to pre-filter the video content and implement an "intelligent" content based retrieval.

When a person (e.g., victim of an aggression) files a complaint, she is asked to describe the elements that could help the human agents find the relevant video segments. The main elements of such description are: the location, the date and time, the victim's trajectory and some distinguishing signs that could be easily noticed in the video (e.g., clothes color, logos). Based on the spatial and temporal information and on their own knowledge concerning the cameras location, the surveillance agents select the cameras that could have filmed the victim's trajectory. Then, the filtered content is visualized in order to find the target scenes, objects (or people) and events.

Based on these observations, the contribution of this paper concerns the video filtering and retrieval. We did an analysis of the current query processing mechanism within the video-surveillance systems that highlighted the fact that the entry point of any query is a trajectory reconstituted based on a person's positions and a time interval. These elements are used to select the videos of the cameras that are likely to have filmed the scenery of interest. Consequently, the video retrieval is treated as a spatio-temporal data modelling problem. In this context, we have proposed the following elements:

- A definition of the hybrid trajectory query concept, trajectory that is constituted of geometrical and symbolic segments represented with regards to different reference systems (e.g., geodesic system, road network);

(c) 2015, Copyright is with the authors. Published in the Proceedings of the BDA 2015 Conference (September 29-October 2, 2015, Ile de Porquerolles, France). Distribution of this paper is permitted under the terms of the Creative Commons license CC-by-nc-nd 4.0.
(c) 2015, Droits restant aux auteurs. Publié dans les actes de la conférence BDA 2015 (29 Septembre-02 Octobre 2015, Ile de Porquerolles, France). Redistribution de cet article autorisée selon les termes de la licence Creative Commons CC-by-nc-nd 4.0.
BDA 2015, 29 septembre au 2 octobre 2015, Ile de Porquerolles, France.
ISSN 2429-4586.

- A multi-layer data model that integrates data of the road network, the transportation network, the objects movement and the cameras' fields of view changes;

- A set of operators that, given a trajectory query and a time interval, select the fixed and mobile cameras whose fields of view are likely to have filmed the query trajectory.

2. RELATED WORK

The video retrieval projects research projects generally focus on developing algorithms based on feature extraction that are exhaustively executed on the available video collections. Very few of them consider a previous video filtering step. In the following we present some of these projects with a focus on content filtering before feature extraction. The CANDELA project proposes a generic distributed architecture for video content analysis and retrieval [7]. The exhaustive content analysis is conducted in a distributed manner at data acquisition using a number of tools. The CARETAKER project [1] investigates techniques allowing automatic extraction of relevant semantic metadata from raw multimedia. Nevertheless, there is no filtering of the content before the feature extraction. More related to our work, the VANAHEIM European project [2], based on the human abnormal activity detection algorithms, proposed a technique for automatically filter (in real time) the videos to display on the video wall screens. Nevertheless, the filtering is based on a video analysis based learning process that supposes the utilization of a big volume of data and that is difficult to implement on a larger scale.

In the following, we present research works aiming at organizing and retrieving visual content based on spatio-temporal information.

[6], proposes a system (SEVA) that annotates each frame of a video with the camera location, the timestamp and the identifiers of the objects that appear in that frame. Therefore this solution can only be applied in a controlled environment. In [8], an approach similar to SEVA is proposed with the following differences: (1) the objects don't have to transmit their positions and (2) their objects geometry is considered and not only their localisation. For each second of the video, two external databases (OpenStreetMaps and GeoDec) are queried in order to extract the objects (e.g., buildings, parks) that are located in the filmed scene. The system doesn't consider spatial queries. [4] is more related to our work and proposes a framework that associates each frame of the video with the geometry of the viewable scene based on metadata collected from GPS and compass sensors. Based on a region query, the framework can return the video sequences that have intersected the video query region. The main difference between their framework and ours is that they don't address the multimedia retrieval process.

3. DATA MODEL

We proposed a model that integrates different types of information: (1) The road Network, (2) The transportation Network, and the objects and sensors that move in this environment (3) Objects and (4) Cameras.

[1]http://cordis.europa.eu/ist/kct/caretaker_synopsis.htm
[2]http://www.vanaheim-project.eu/

$$hasSeen : u_1, u_2, ..., u_n, [t_1, t_2] => \begin{cases} c_1 : t^1_{start} -> t^1_{end}, u_k (1 <= k <= n) \\ c_2 : t^2_{start} -> t^2_{end}, u_k (1 <= k <= n) \\ ... \\ c_m : t^m_{start} -> t^m_{end}, u_k (1 <= k <= n) \end{cases}$$

Figure 1: The specification of the proposed operator

Definition 1: A road network is a non directed graph G_R = (E,V) where E = $\{e_i$ / e_i=$(v_j$, $v_k)\}$ is a set of road segments and V=$\{v_i\}$ is the set of segments junctions [5].

Definition 2: A transportation network $G_T = (E_T$,V_T) is a non directed graph where $V_T = v_{ti}$ is the set of bus station and $E_T = e_{ti}$ / e_{ti}=$(v_{tj}$, $v_{tk})$ is a set of transportation network sections.

Definition 3: Let MO=$\{mo_i\}$ be the set of mobile object. Let TR(mo_i) be the function that extracts the mobile object's mo_i trajctory. Let $\{position_j(mo_i)\}$ be the set of mobile object's mo_i positions. Let $\{time_j(mo_i)\}$ be the mobile object's mo_i set of timestamps.

Definition 4: Let FC={fc}/ fc is a fixed camera, id(fc)=c_i gives the camera's id, position(c_i) gives the camera's position and fov(c_i) extracts the set of it's field of view changes.

Definition 5: Let MC = {mc} / mc is a mobile camera, id(mc)=c_i gives the camera's id, mo(c_i)=mo_i∈MO extracts the mobile object that the camera is attached to. The camera's trajectory will be the mobile object's one: TR(c_i)= TR(mo(c_i))).

We define two types of positions: *a geometric position* that is a 2D position relative to the geodesic system (GPS <lat, long> coordinates) and *a symbolic position* relative to the underlying layers. We have defined mapping functions that do the connection between the different layers (e.g., compute the position of a bus station or map an object's trajectory with regards to the road network).

Based on the data model, we define the operator *hasSeen* that has as input the query defined as a sequence of spatial segments (u_1, u_2,....,u_n) and a time interval [t_1,t_2]. The result is a list of cameras likely to have filmed the query's trajectory with the corresponding time intervals. The specification of the operator is illustrated in figure 1.

4. THE PROPOSED VIDEOSURVEILLANCE FRAMEWORK

The Figure 3 illustrates the framework we are proposing in two steps: (1) the spatio-temporal filtering (red workflow in the Figure 3) and (2) the multimedia querying (green workflow in the Figure 3). Let's use the query illustrated in 2 as a running example.

Location : Paris
Date and Time : January 23rd 2014 between 10h and 12h
Trajectory : Rivoli Street : Louvre Museum exit -> Subway Chatelet entrance
Description : man dressed in red

Figure 2: Query example

4.1 Spatio-temporal filtering

Query Interpreter is the module that is "translating" the spatial and temporal information given by the user into a spatio-temporal query.

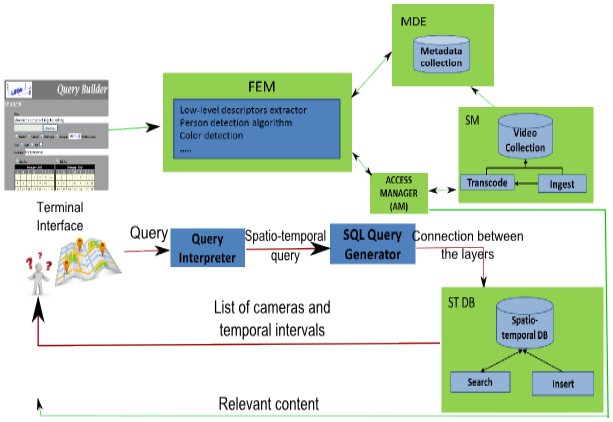

Figure 3: The architecture of the proposed framework

SQL Query Generator is the module that based on the spatio-temporal query implements the algorithms 1 and 2. Here are the used methods:

extractCamDist(u_k, max(FOV.visibleDistance)): fixed cameras filtering with regards to the query segments and the maximum visible distance of the cameras in the database.

geometries computation and intersection: compute cameras fields of view geometries and generate SQL queries for intersection with the queries segments; the queries are then executed on the **Spatio-temporal database**.

The schema from Figure 4 illustrates a road network (S1-S5 and S6-S10). The fixed cameras (C_1, C_2, C_3) positions and fields of view are shown. Suppose the query trajectory is TR = S_1,S_2,S_3,S_4,S_5) (Rivoli Street: Louvre Museum exit -> Subway Chatelet entrance) and the time interval [t_1, t_2] (January 23rd 2014 between 10h and 12h).

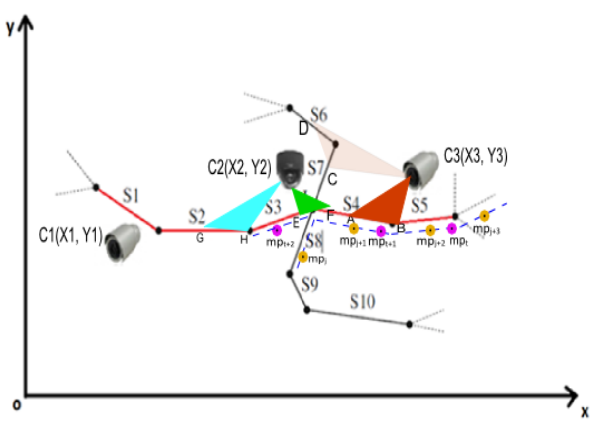

Figure 4: A road network filmed by three fixed cameras

The Figure 5 illustrates the different fields of view of the cameras C_2 and C_3 in time (fov(C_2) and fov(C_3)). The different moments when the fields of view change are marked with colors corresponding to the geometries from the Figure 4 (e.g., at $time_j(fov(C_3))$ the field of view becomes ABC_3).

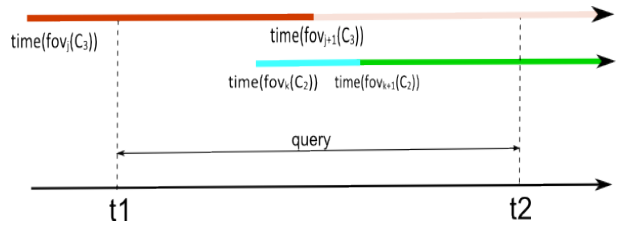

Figure 5: The moments when the fields of view change and the query interval

The first lines of the algorithm 1 (1-3) represent a filtering step. From all the cameras in the database we will select only those located at a distance smaller than the maximum visible distance from the database. In our case the only cameras that have possibly filmed the query's trajectory segments are C_1, C_2 et C_3.

Algorithm 1: Fixed cameras selection

```
1  for each u_k of the query do
2      camList ←
       extractCamDist(u_k, max(visibleDistance))
3  end
4  for each c_i from camList do
5      for each (fov_j(c_i)) do
6          if time(fov_j(c_i)) >=t_1 and time(fov_j(c_i))<=
           t_2 then
7              geometry_ij ← construct_polygon(fov_j(c_i));
8              for each u_k of the query do
9                  if geometry_ij intersects u_k then
10                     add(c_i, u_k, [time(fov_j),
11                     min(succ(time(fov_j)), t_2)]);
12                 end
13             end
14         end
15         if time(fov_j(c_i)) < t_1 and
           t_1<=time(succ(fov_j(c_i))) then
16             geometry_ij ← construct_polygon(fov_j(c_i));
17             for each u_k of the query do
18                 if geometry_ij intersects u_k then
19                     add(c_i, u_k, [t_1, min(time(succ(fov_j)), t_2)]);
20                 end
21             end
22         end
23     end
24 end
```

For each camera selected at the first step, we will search the periods with changes in the field of view (lines 4,5 of the algorithm 1). The lines 6-19 process the two possible cases: the change is between t_1 and t_2 (e.g., time($fov_k(C_2)$) or the change is before t_1 (e.g., time($fov_j(C_3)$)). The geometries are build and the intersection with the query's trajectory is evaluated.

The result is:
{(C_2, S_2, [time($fov_k(C_2)$), time($fov_{k+1}(C_2)$)]), (C_2, S_3, [time($fov_{k+1}(C_2)$), t_2])), (C_2, S_4, [time($fov_{k+1}(C_2)$), t_2))), (C_3, S_4, [t_1, time($fov_{j+1}(C_3)$)])}.

We now consider two mobile objects which trajectories are represented as dotted lines on the figure 4. By mobile object we understand any entity capable of transmitting a periodically update of its position. Lets suppose that each object sends at least one update mp_j (mobile position) containing its position and a timestamp per road segment. By considering each road segment and each mobile object (lines 1-2 of the algorithm 2), the function $filter(mo_i, u_k, [t_1, t_2])$ will test the possible cases: the object's position is on the query's trajectory between t_1 and t_2 (e.g., mp_t, mp_t, mp_{j+1}, mp_{j+2} like illustrated in Figure 6) and the preceding position intersects also (mp_{j+1} and mp_{j+2}) or the preceding position doesn't intersects the trajectory (mp_j and mp_{J+1}) or it intersects but before t_1 (mp_t and mp_{t+1}).

The result is: $\{(obj_i,\ S_4,\ [t_1, \text{time}(mp_{j+1})]),\ (obj_i,\ S_5,\ [\text{time}(mp_{j+1}),\ t_2]),\ (obj_{i+1},\ S_4,\ [\text{time}(mp_t), t_2])]\}$

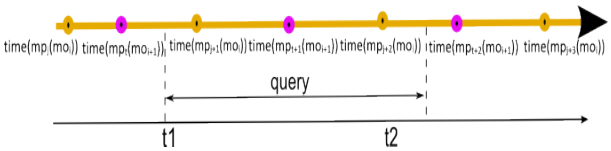

Figure 6: The mobile object's trajectory points and the query interval

Algorithm 2: Mobile cameras selection

```
1 for each u_k do
2     for each mo_i do
3         listMobileObj ← add(filter(mo_i, u_k, [t_1, t_2]));
4     end
5 end
6 for each mo_i.id from listeObjMobiles do
7     listeCameras ← selectionnerCameras(mo_i.id);
8 end
```

4.2 The multimedia retrieval

Once the spatio-temporal filtering is done, the video content is analyzed based on the multimedia query engine. Two types of inputs are allowed: (1) textual query (e.g., people dressed in red etc.) and (2) image query. This search is iterative so for our query example we have the next scenario. The victim remembers that the aggressor was wearing a red coat. The tool that detects people and the main color of their upper body is executed and the first set of results is presented to the user. He watches them and selects a new image query. The image that allowed identification was the one illustrated in the left part of the figure 8.

The LINDO project defined a generic and scalable distributed architecture for multimedia content indexing and retrieval. We used the components of the Video Surveillance server from Paris (described in [1]).

The Access Manager (AM) provides methods for accessing the multimedia contents stored into the **Storage Manager**. The method the most received from the FEM is String *extract(String track, long beginTime, long endTime)*: starts the processing of a track between the time beginTime and the time endTime.

The Feature Extractors Manager (FEM) is in charge of managing and executing a set of content analysis tools over the acquired multimedia contents. It can permanently run the tools over all the acquired contents or it can execute them on demand only on certain multimedia contents. The FEM implementation is based on the OSGI framework [3], the tools or extractors are exported as services and any algorithm that respects the input and output interfaces can be integrated. In our implementation we used tools developed by two of the project's partners [4], [5] and that are illustrated in figure 8.

The Metadata Engine (MDE) collects all extracted metadata about multimedia contents. In the case of a textual query, the metadata can be queried in order to retrieve some desired information. The metadata is stored in an XML format presented in [2].

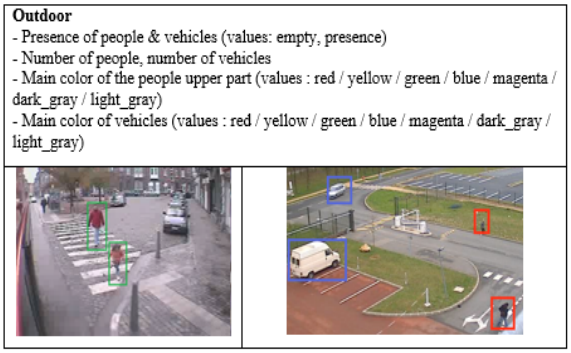

Figure 7: Illustration of the content analysis tools

```
<document src="stream1">
 <video capturedBy="cam1_Paris">
  <object type="Person" id="0">
   <localisation confidence="100">
    <period start_time="2010-07-28T11:07:35" end_time="2010-07-28T11:07:55"/>
    <area>parking area entry A2</area>
   </localisation>
   <property name="color">red</property>
  </object>
 </video>
</document>
```

Figure 8: Example of metadata generated by the color detection tool

5. CONCLUSIONS

We presented in this paper a video retrieval framework that has two main components: (1) a spatio-temporal filtering module and (2) a content based retrieval module (based on a generic framework for indexing large scale distributed multimedia contents that we have developed in the LINDO project).

The generic architecture aims to guide the design of systems that could assist the video surveillance operators in their research. Starting from a sequence of trajectory segments and a temporal interval, such system generates the

[3] http://www.osgi.org/Main/HomePage
[4] http://www.supelec.fr/
[5] http://www-list.cea.fr/

list of cameras that could contain relevant information concerning the query (that 'saw' the query's trajectory) then executes some content analysis tools that could automatically detect objects or events in the video.

For now, our model considers only outdoor transportation and surveillance networks. We plan to extend our model to indoor spaces also in order to model cameras inside train or subway stations for example.

6. REFERENCES

[1] M. Brut, D. Codreanu, S. Dumitrescu, A.-M. Manzat, and F. Sedes. A distributed architecture for flexible multimedia management and retrieval. In *Proceedings of the 22nd International Conf. on Database and Expert Systems Applications*, DEXA'11, pages 249–263, 2011.

[2] M. Brut, S. Laborie, A. Manzat, and F. Sedes. A generic metadata framework for the indexation and the management of distributed multimedia contents. In *3rd International Conf. on New Technologies, Mobility and Security (NTMS)*, pages 1–5, Dec 2009.

[3] R. Cucchiara. Multimedia surveillance systems. In *Proceedings of the Third ACM International Workshop on Video Surveillance and Sensor Networks*, VSSN '05, pages 3–10. ACM, 2005.

[4] B. Epshtein, E. Ofek, Y. Wexler, and P. Zhang. Hierarchical photo organization using geo-relevance. In *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*, GIS '07, pages 1–7, 2007.

[5] K. Liu, Y. Li, F. He, J. Xu, and Z. Ding. Effective map-matching on the most simplified road network. *In Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 609–612, 2012.

[6] X. Liu, M. Corner, and P. Shenoy. Seva: Sensor-enhanced video annotation. *ACM Trans. Multimedia Comput. Commun. Appl.*, pages 1–26, 2009.

[7] P. Merkus, X. Desurmont, E. G. T. Jaspers, R. G. J. Wijnhoven, O. Caignart, J. f Delaigle, and W. Favoreel. Candela - integrated storage, analysis and distribution of video content for intelligent information systems, 2004.

[8] Z. Shen, S. Arslan Ay, S. H. Kim, and R. Zimmermann. Automatic tag generation and ranking for sensor-rich outdoor videos. In *Proceedings of the 19th ACM International Conf. on Multimedia*, MM '11, pages 93–102, 2011.

Bi-temporal Query Optimization Techniques in Decision Insight

Azhar Ait Ouassarah[1,2]
Xavier Fournet[1]
Romain Revol[1]

[1] Axway
171 Rue Bureaux de la Colline
92210, St Cloud, France
{aaitouassarah, naverseng, xfournet, rrevol}@axway.com

Nicolas Averseng[1]
Jean-Marc Petit[2]
Vasile-Marian Scuturici[2]

[2] Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205
20 avenue Albert Einstein
69100, Villeurbanne, France
{azhar.ait-ouassarah, jean-marc.petit, vasile-marian.scuturici}@insa-lyon.fr

ABSTRACT

La complexité de l'environnement dynamique dans lequel évoluent les entreprises requiert de la part de leurs managers de prendre des décisions pertinente en un laps de temps très court. Les systèmes de *supervision des activités métiers* devraient supporter des requêtes bitemporelles complexes qui accèdent aussi bien à des données historiques qu'à des données temps réel afin de détecter des anomalies ou bien des tendances dans l'activité de l'entreprise. Cependant, il s'avère que l'accès à ces deux types de données peut être lent, ce qui ne convient pas aux applications de supervision.
Dans ce papier, nous présentons Decision Insight, une plateforme développée par un éditeur de logiciels français pour aborder ce problème. Elle est basée sur un SGBD orienté colonnes qui redéfinit les requêtes bi-temporelles en: 1) un ensemble de requêtes continues pour gérer les données temps réel et dont les résultats sont matérialisés, et 2) une requête qui accède aussi bien aux données historiques qu'aux résultats des requêtes continues.
Nous démontrons l'intérêt de notre approche en utilisant une version adaptée du benchmark TPC-BiH qui est une extension bitemporelle du benchmark TPC-H.

Categories and Subject Descriptors

H.4 [**Information Systems**]: Data management systems—*Data model extensions,Temporal data*

General Terms

Business activity monitoring, temporal databases, temporal query optimization

(c) 2015, Copyright is with the authors. Published in the Proceedings of the BDA 2015 Conference (September 29-October 2, 2015, Ile de Porquerolles, France). Distribution of this paper is permitted under the terms of the Creative Commons license CC-by-nc-nd 4.0. ISSN 2429-4586.
(c) 2015, Droits restant aux auteurs. Publié dans les actes de la conférence BDA 2015 (29 Septembre-02 Octobre 2015, Ile de Porquerolles, France). Redistribution de cet article autorisée selon les termes de la licence Creative Commons CC-by-nc-nd 4.0. ISSN 2429-4586.
BDA 29 septembre 2015, Ile de Porquerolles, France.

Keywords

Historical data, real-time data, materialized views, data reduction

1. INTRODUCTION

Companies operate in very dynamic and complex environments, requiring their managers to possess both agility and ability to make proactive operational decisions, in order to maintain or improve their business [23]. On the one hand, exploiting historical data is covered by the Business Intelligence (BI) domain [33], which provides access to past performance indicators by analyzing information stored in data warehouses. This enables users to understand what happened in the past and help them to prevent making mistakes in the future.
On the other hand, managers traditionally rely on Business Activity Monitoring (BAM) systems [23] to make operational decisions, see for instance `Splunk`[1] and `Vitria`[2]. BAM aims to provide real-time access to critical business performance indicators. Thus managers can have a deep insight into what is currently happening in their business before taking rapid and effective decisions. BAM gathers its information in real-time by analyzing data streams from multiple sources. Nevertheless these technologies are limited because they focus on real-time information, ignoring existing historical data. They do not give managers the necessary hindsight to compare the current company activity behavior with its history. Therefore, false positive decisions could be taken by analysts since real-time and historical data are not considered together.
To overcome these limitations, BAM tool capabilities could be enhanced with BI features. Managers could then use an hybrid tool to handle both real-time and historical data, allowing them to continually navigate from real-time to historical data. Such hybrid systems approach faces with two main issues: *combining real-time and historical systems* and *the performance issue*.

- *Handling Two Different Types of Systems*: Historical and real-time data are handled by two different types of systems. Historical data require well-known OLAP DataBase Management Systems (DBMS). Real-time data rely on Data Stream Management Systems (DSMS), Complex Event Processing

[1] `http://www.splunk.com/`
[2] `http://www.vitria.com/`

(CEP) or BAM systems [22]. These systems handle transient data streams and can query them thanks to *Continuous Queries*.
Handling within a unique system those two different types of data is clearly an issue.

- *Performance Issue*: BI systems are usually used to generate non interactive reports which do not have real-time (or near real-time) requirements. BAM tools however always provide managers with a graphical user interface (GUI) to monitor their business. The GUI is always interactive and offers the possibility to explore real-time data and other analyses computed over them. This means that the underlying system must guarantee fast response time of queries in charge of feeding the GUI with information. This is because GUI display lag can make the system unpractical, thus reducing its interest.
When historical data have to be queried, the main performance bottleneck is on the induced I/O cost which impacts real-time data processing, leading to unacceptable GUI display lag.

As far as we know, Chandrasekran and Franklin [9] and Reiss and al [24] were the first to address the topic of enhancing DSMS with DBMS capabilities. In [24], they define bitmap index specially designed to handle these two types of data. In [9], they present a framework using Data Reduction techniques, sampling techniques in their case, to limit I/O overhead induced by accessing historical data. Their approach enables the reduction level to be adapted according to the available resources. They also addressed the issue of when to perform data reduction. Three approaches were explored: data reduction at data arrival, at query execution or both. The framework was implemented on top of PostgreSQL.

In this paper, we focus on queries that access both historical data already existing in the system and live data that had not yet been entered into the system when the query was defined. They are temporal by nature because they access data evolving over time. Such queries support either the *valid time dimension* [28] or the *transaction time dimension* [29] or both (*bi-temporal query*). The support of the *transaction time* means that the query can access the history of data as it is modified in the DB. The *valid time* enables access to the history of data as it evolves in the modeled reality. We consider the case of bi-temporal queries. Let us consider the following example:

> "*What is the average of new revenue achieved by the company every month last year (2014), considering the DB at the instant 1/10/2015 ?*"

This simple query requires access to all items that have been ordered from 1/1/2014 to 12/31/2014. Depending on the database, this query may induce a large number of I/O operations and can be very long to complete. Whenever this query is executed only once, traditional approaches can be applied. Otherwise, if this query has to be executed several times – for example to frequently refresh some GUI – this is not acceptable.

Paper contribution.

We present Decision Insight [3], a platform that deals with historical and real-time data in a unified manner. The project was launched in 2008 under the name of Tornado by Systar, a French software editor. This latter was then acquired in 2014 by Axway, another French software editor and Tornado became Decision Insight. Axway aims to compete with major leading industries in data management worldwide and turns out be the fifth french editor according to a recent study[3]. The project mobilizes about twenty engineers and it has been marketed since 2013. This platform implements an optimization that consists in redefining complex bi-temporal queries into: 1) a set of continuous queries in charge of handling real time data streams (whose results are materialized) and 2) a query that accesses both historical and materialized results of the previous continuous queries. Thus, Decision Insight can provide analysts with timely answers through a convenient GUI [3].
Decision Insight is based on a column-store bi-temporal DBMS that handles these two types of data and implements a simple and efficient bi-temporal query optimization technique. We demonstrate the interest of our approach using an adapted version of TPC-BiH, a bi-temporal extension of the TPC-H benchmark. Extensive experiments have been conducted, pointing out the interest of Decision Insight for delivering timely information based on historical and real-time data.

Paper organization.

The remainder of this paper is structured as follows: In section 2 we introduce our query rewriting approach. Then in section 3, we address the issue of materialized continuous query computation scheduling. Then in section 4, we point out how the contribution has been implemented within the Decision Insight framework. Experiments are given in section 5 using the TPC-BiH [17] benchmark. Section 6 is devoted to related works. Then we conclude the paper in section 7.

2. QUERY REWRITING

In this section, we first describe how aggregate bi-temporal SQL queries are expressed. Then we explain how complex bi-temporal queries are decomposed as a set of materialized continuous queries (CQ) and one special bi-temporal query.

In the rest of the paper, all queries are expressed using a pseudo-SQL formalism based on SQL:2011 supporting temporal features [20].

2.1 Temporal Query Expression for Aggregate Queries

It is well known that the expressiveness of SQL:2011 is limited to defining temporal queries performing aggregations [20, 18].

2.1.1 Rhythm

To deal with this issue, we introduce the concept of *rhythm* as a partition of the valid time domain into contiguous and equal-length time intervals. A rhythm is defined by a couple $(begin, duration)$ where $begin$ is the reference time instant to be used for partitioning the valid time domain and $duration$ is the length of each interval. For example, the rhythm $(01/01/2014, 1\text{ day})$ corresponds to the following partition:

$$[01/01/2014, 01/02/2014[\cup[01/02/2014, 01/03/2014[\cup \ldots$$

Clearly, a rhythm is used to define the time range over which the aggregation is performed.

Rhythms can be implemented in SQL:2011 as a relation with two attributes representing the endpoints of each interval of that rhythm. In the sequel, all queries use a one-day rhythm $(01/01/1990, 1\text{ day})$ represented by the relation $Rhythm_day$.

2.1.2 Temporal Data Schema of the Running Example

[3]http://www.infodsi.com/articles/155700/editeurs-francais-contre-courant-conjoncture.html?key=c9fee7a30353fd4a

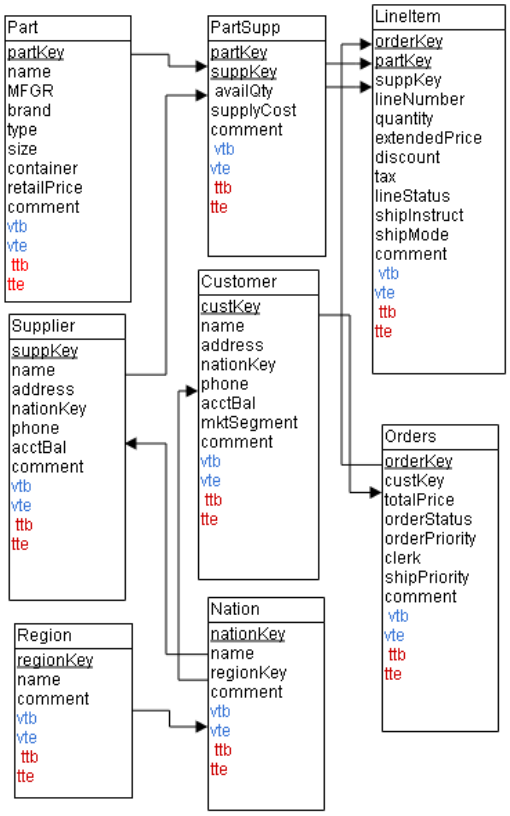

Figure 1: Temporal database schema adapted from TPC-BiH [17]

The TPC-BiH database schema [17] is a bi-temporal extension of the TPC-H database schema. It mainly describes customers, orders and line items of a fictive company. We have slightly adapted the TPC-BiH database schema to fit our needs (Fig. 1). The schema is fully bi-temporal, i.e all tables are extended with both valid time and transaction time.

2.1.3 *Temporal Aggregations*

We follow Kaufmann and al's classification of time ranges in temporal aggregations [18]. They identified four types of time ranges:

- **Instantaneous Aggregation** [12]: The aggregation is performed on all valid tuples at an instant, e.g "What is the number of orders with orderstatus='O' at the beginning of every day during the valid time interval [01/01/1994, 06/01/1994[, considering the DB at the transaction instant '10/20/2014T10:30:00.0' ?". This query is represented in Listing 1.

Listing 1: Example of an instantaneous aggregation

```
SELECT COUNT(*) as numberOpenOrders,
  Rhythm_day.vtb, Rhythm_day.vte
FROM Orders
FOR SYSTEM_TIME AS OF TIMESTAMP
'10/20/2014T10:30:00.0' ,
  Rhythm_day
WHERE
  -- filter of rhythm intervals
  (01/01/1994 <= Rhythm_day.vtb AND
  Rhythm_day.vtb < 06/01/1994) AND
  -- only orders valid at Rhythm_day.vtb
  Orders.vtb <= Rhythm_day.vtb AND
  Rhythm_day.vtb < Orders.vte  AND
  orderstatus= 'O'
GROUP BY Rhythm_day.vtb;
```

- **Tumbling Window**: the aggregation is performed on non overlapping intervals, e.g "What is the total revenue achieved by the company every day during the valid time interval [01/01/-1994, 06/01/1994[and considering the DB at tt=10/20/2014T-10:30:00.0 ?". This query is represented in Listing. 2.

Listing 2: Tumbling Window query

```
SELECT SUM(extendedPrice) as totalRevenue,
  Rhythm_day.vtb, Rhythm_day.vte
FROM LineItems
FOR SYSTEM_TIME AS OF TIMESTAMP
'10/20/2014T10:30:00.0',
  Rhythm_day
WHERE
  -- filter of rhythm intervals
  (01/01/1994 <= Rhythm_day.vtb AND
  Rhythm_day.vtb < 06/01/1994) AND
  -- filter new lineItems
  Rhythm_day.vtb <= LineItems.vtb AND
  LineItems.vtb < Rhythm_day.vte
GROUP BY Rhythm_day.vtb;
```

- **Sliding Window**: the aggregation is performed on overlapping intervals, e.g "What is the total revenue achieved by the company during the last 10 days, computed every day if we consider the valid time interval [01/01/1994, 06/01/1994[and considering the DB at tt = '10/20/2014T10:30:00.0' ? ". This query is represented in Listing 3.

Listing 3: Aggregation using a sliding window time range

```
SELECT SUM(extendedPrice) as totalRevenue,
  Rhythm_day.vtb, Rhythm_day.vte
FROM LineItems
FOR SYSTEM_TIME AS OF TIMESTAMP
'10/20/2014T10:30:00.0',
  Rhythm_day
WHERE
  -- filter of rhythm intervals
( 01/01/1994 <= Rhythm_day.vtb AND
  Rhythm_day.vtb < 06/01/1994) AND
  -- filter new lineItems
  (Rhythm_day.vtb - INTERVAL 10 days) <=
  LineItems.vtb AND
  LineItems.vtb < Rhythm_day.vtb
GROUP BY Rhythm_day.vtb;
```

- **Landmark Window**: the aggregation is performed on intervals that share the same interval begin, e.g "What is the total revenue achieved by the company to each day for the current month considering the valid time instant [01/01/1994, 01/23/1994[and the DB at tt = '10/20/2014T10:30:00.0' ?". This query is represented in 4.

Listing 4: Aggregation using a landmark window time range

```
SELECT SUM(extendedPrice) as totalRevenue,
```

```
    '01/01/1994' as vtb, Rhythm_day.vtb

  FROM LineItems
  FOR SYSTEM_TIME AS OF TIMESTAMP
  '10/20/2014T10:30:00.0',
    Rhythm_day
  WHERE
    -- filter of rhythm intervals
    (01/01/1994 <= Rhythm_day.vtb AND
    Rhythm_day.vtb < 01/23/1994) AND
    -- filter new lineItems
    01/01/1994 <= LineItems.vtb AND
    LineItems.vtb < Rhythm_day.vtb
  GROUP BY Rhythm_day.vtb;
```

2.2 Query Rewriting Technique

Business Activity Monitoring systems usually provide managers with features to build so-called views to monitor their business through user-friendly GUI. Those views use underlying queries to feed them with information to display. Consequently, they are not intended to be executed only once and then deleted as is usually the case in standard applications. Indeed, they can be evaluated several times, as long as the related view needs to be updated.

In this section, we sketch the main idea of our query rewriting technique. Without loss of generality, we are concerned with the following class of bi-temporal queries [30]:

Listing 5: Initial query Q_t

```
SELECT A1, A2, ..., An, Agg1, Agg2, ..., Aggk,
  Rhythm_Table.vtb,Rhythm_Table.vte
FROM table1, table2, ..., tableJ,
  stream1, stream2, ..., streamK,
  Rhythm_Table
WHERE tc1 AND tc2 AND ... AND tcn AND
  c1 AND c2 AND ... AND cm
GROUP BY A1, A2, ..., An,
  Rhythm_Table.vtb, Rhythm_Table.vte
```

where:

- $A_1, A_2, \ldots, A_n$ are attributes or derived attributes,
- Agg_1, Agg_2, ..., Agg_k are aggregation functions, e.g., AVG, SUM, MIN.
- The WHERE clause is a conjunction of selection predicates and join predicates: tcj predicates are over temporal attributes while ci are over non temporal ones.
- $table_1, table_2, \ldots, table_J$ are tables from the accessed database (historical data).
- $stream_1$, $stream_2$, $\ldots$, $stream_K$ are data streams (live data)
- $Rhythm_Table$ is the table defined in the previous section.

Such a query is used to feed an underlying GUI whenever it needs to be updated, e.g due to a user interaction. Whenever the amount of data to be processed exceeds some limits, GUI latency deteriorates. Therefore, to address the scalability issue, we rely on data reduction techniques. Intuitively, we compute as soon as possible some partial answers allowing to efficiently answer a query asked by decision-makers. In other words, instead of performing aggregations at query time, we propose to perform them on data arrival. Thus, when a query is executed, it simply accesses the results of the aggregations which requires fewer I/O operations. This approach ensures that the most expensive I/O costs have been performed before information is needed by a decision-maker. Hence, at query-time, the cost will be as low as possible, thus satisfying our major goal.

Given a bi-temporal query, the process is as follows:

- one or more simple *continuous queries* [4] are defined, and their results are materialized. Such queries handle large volumes of data and do not affect historical data. They are referred to as *materialized continuous queries*;
- one elaborated temporal query, referred to as an *on-demand query*, in charge of providing decision-makers with results. Such a query accesses both historical and live data, including materialized CQs.

This approach has the advantage of providing a unified way to access both real-time and historical information through temporal queries. The result of this approach is *equivalent* to the result of the initial query against the same data. The reader is referred to [19] for equivalence of continuous queries. This is not in the scope of this paper.

2.3 Materialized Continuous Queries

For each aggregation Agg_i in the initial query, we define one continuous query in charge of reducing input data into pre-computed aggregates. This query is simple and can handle a large volume of data, as in (Listing 6).

Listing 6: A materialized continuous query

```
SELECT A1,  A2, ..., An, Agg, vtb, vte
FROM stream1, stream2, ..., streamK
WHERE tc1 AND tc2 AND ... AND tcn AND
  c1 AND c2 AND ... AND cm
GROUP BY A1,  A2, ..., An
```

where:

- Agg is the aggregation operation performed by the query,
- $stream1, stream2, ..., streamK$ is the set of accessed data streams,
- vtb and vte are two time attributes representing the time interval during which the computed result is valid,
- the result of this query is stored in a table, thus becoming historical data.

Each continuous query is bound at its creation to a *rhythm*. For each interval of the rhythm, the query returns one result that is stored in the DB. The choice of the rhythm depends on the user's needs. The more accurate the expected result, the finer the rhythm's granularity, and the higher CPU cost and memory utilization.

Whenever a continuous query is created, some new attributes linked to that query are added dynamically to the database schema. This is intended to store the query results for future use. If we consider the example from the introduction, then our approach requires one continuous query (Listing. 7).

Listing 7: The continuous query sumNewRevenuePerDay

```
SELECT SUM(extendedPrice) as agg,
  [vtDay].vtb, [vtDay].vte
FROM LineItems
WHERE
  [vtDay].vtb <= vtb AND
```

```
  vtb < [vtDay].vte) AND
  Rhythm_day.vtb <= LineItems.vtb AND
  LineItems.vtb < Rhythm_day.vte;
```

2.4 On-demand Queries

An on-demand query is a bi-temporal query executed against the database whenever new information is required by decision-makers through their GUI.

Listing 8: On-demand query

```
SELECT A1, A2, ..., An
FROM table1, table2, ..., tableJ
WHERE tc1 AND tc2 AND ... AND tcn AND
  c1 AND c2 AND ... AND cm;
```

We consider two classes of *on-demand queries*: *time travel* and *time slice* queries.

- **Time Travel**: This consists in acquiring the state, or snapshot, of the DB at a specific time. Here is an example of this class of queries (listing 9): " What is the revenue achieved by the company at vt = 10/13/2014, considering the DB at tt = 10/20/2014T10:30:00.0?".

Listing 9: A time travel query example

```
SELECT aggr
FROM sumNewRevenuePerDay
FOR SYSTEM_TIME AS OF TIMESTAMP
10/20/2014T10:30:00.0
WHERE vtb <= 10/13/2014 AND 10/13/2014 < vte;
```

- **Time Slice**: This class of queries is intended to return the historical data according to one temporal dimension. We fix one time dimension at a particular time instant while the other one is fixed at an interval. Here is an example of this class of queries (listing 10): "What is the revenue achieved by the company during the interval [10/13/2014, 12/13/2014[and considering the DB at tt=10/20/2014T10:30:00.0?"

Listing 10: A time slice query example

```
SELECT aggr, vtb, vte
FROM sumNewRevenuePerDay
FOR SYSTEM_TIME AS OF TIMESTAMP
10/20/2014T10:30:00.0
WHERE 10/13/2014 <= vtb AND vtb < 12/13/2014;
```

2.5 Data Storage

A database used in BAM applications has to store both real-time data and results of the materialized continuous queries. It thus offers an unified interface to access them all. Since these data are bi-temporal, we need a database management system with bi-temporal capabilities. Decision Insight is based on a column-store DBMS [31, 7] which means that a table is stored column by column. We outline below three main reasons motivating our choice.

2.5.1 Performances in Analytical Workloads

The *oriented-column* databases are intended to perform analytical queries that analyze data and give an insight into the business activity, e.g the number of orders in pending status. The column-oriented database systems outperform row-oriented database systems on analytical workloads such as those found in business intelligence and decision support applications [1].

2.5.2 Dynamic Update of the Database Schema

Our query optimization requires adding and removing attributes dynamically. The row-oriented approach is not suitable in our case because addition or deletion of an attribute affects the whole table, with performance impacts on the modified table. However the column-oriented approach does not suffer from this issue since each attribute has its own column.

2.5.3 Dealing with Temporal Attribute Evolution

The database must have bi-temporal built-in support and is intended to handle the history of data as it evolves. In the row-oriented approach, the update of an attribute value requires adding a new tuple with the new value. This behavior causes both a storage overhead and an increase in query execution time due to data duplication. As an example, let us consider the table *Customer* (Table 1) represented using the formalism proposed by Snodgrass [27]. The update of the attribute *balance* for the customer "AWM" leads to the insertion of two new tuples in the table 2.
In a column-oriented approach, each attribute can evolve independently because it is stored in its own column.

Table 1: customer before the update

custid	name	balance	vtb	vte	ttb	tte
1	Benason	12000	10/16	∞	10/16	∞
2	AWM	2000	10/9	∞	10/10	∞
3	Vop	47800	10/13	∞	10/14	∞

Table 2: customer after the update

custid	name	balance	vtb	vte	ttb	tte
1	Benason	12000	10/16	∞	10/16	∞
2	AWM	2000	10/9	∞	10/10	10/17
2	AWM	2000	10/9	10/17	10/17	∞
2	AWM	6000	10/17	∞	10/17	∞
3	Vop	47800	10/13	∞	10/14	∞

This issue has been addressed by Jensen and Snodgrass from a logical point of view [16]. They support the idea that the introduction of temporal dimensions in the data model requires adaptation of the data model design. They propose a handbook of best practices for the design process. However, this is not in the scope of this paper. As shown above, Decision Insight is therefore based on a column-store DBMS.

2.6 Computation Scheduling of Materialized Continuous Queries

The use of materialized views requires in general to consider the scheduling strategy to compute its results. This strategy has to reconcile keeping views up-to-date as data is collected and limit the number of refresh so the computation cost overhead is contained. Actually it is unthinkable to compute a view for each single incoming update of entity. It is rather wise to refresh views periodically or by bunch of updates. In the case of a soft real-time 2TDBMS, the maintenance of materialized queries is more complex. The real-time aspect induces computation deadline constraints to ensure fresh information to managers. The bitemporality necessitates to consider the semantics of the two temporal dimensions to choose the adapted computation strategy. Indeed we need to determine the adequate instant when data is supposed to be available in the database to trigger the computation. In the general case the two dimensions are orthogonal, which means that there is no restrictions

between the valid time and the transaction time of any fact in the DB. However in many practical applications there is a restriction relationship between them. For example, if we suppose that every event that occurs in the reality is considered as valid when it is inserted in the DB which, then $vt_e = tt_e$. This topic has been addressed by Jensen and Snodgrass in bitemporal relational databases [15] under the name *temporal specialization relations*. The authors classify bitemporal relations into 15 classes of specialization.
In Decision Insight, we consider three types of events :

- *Retroactively bounded events*: It is the usual case. For each event, valid time and transaction time have the following interrelationships $vt_e < tt_e \leq \Delta t + vt_e$ with $\Delta t > 0$. In specific terms, the event occurs in reality at vt_e, then it is recorded in DB at tt_e. Δt is fixed by the user and represent the necessary time to collect it, transfer it to the DB and record it.
- *Delayed retroactive events*: It corresponds to events whose temporal attributes have the following interrelationships $\Delta t < tt_e - vt_e$. This type of events occurs in two cases:1) when there is technical issue making difficult to deliver events to the DB. 2) to correct previous events that have been recorded into the DB.
- *Predictive events*: This case corresponds to events that are recorded in the DB before they occurs in reality ($tt_e \leq vt_e$), e.g a government tax rate modification which is always announced before it is applied so that concerned people make arrangements.

In order to handle these three types of events, Decision Insight implements two different approaches: *Live Mode* and *Late Data Handler*.

- *Live Mode* : This approach is the usual mode and is in charge of handling both *retroactively bounded events* and *predictive events*. Concretely, considering a *materialized continuous query*, the condition to schedule its execution for an interval of its rhythm is that all input data are available. Thus, for a rhythm interval $[vt_{begin}, vt_{end}[$, the system supposes that at $tt = vt_{end} + \Delta t$ all input data is available and schedules the computation. Δt must smaller than $vt_{end} - vt_{begin}$. Otherwise, the computation task queue fill rate will be faster than the computation rate.
- *Late Data Handler*: This mode is dedicated to *retroactive events*. When such type of events arrives, the system determines all *materialized continuous query* and *rhythm intervals* impacted. Then it schedules a their recomputation.

In the sequel, we restrict ourselves to the *live mode*.

3. DECISION INSIGHT

Decision Insight is a comprehensive data-intensive decision support system that combines both BI and BAM capabilities. It uses a dashboard as an user interface primitive, allowing analysts to visualize activity indicators and to navigate in time to understand how they evolve. A dashboard is made up of one or more graphical elements (diagrams, charts, datagrids, ...) referred to as *pagelets* in the sequel. Each graphical element displays data returned by an underlying query. An example of a pagelet is depicted in Fig. 2, where daily revenues for a company are displayed as a curve in a particular time range.

Implementing queries using a SQL-based language can be a very difficult activity, particularly for business managers with limited technical skills. Decision Insight provides an advanced graphical interface for rapid design of the complex queries related to BAM [3].
Decision Insight allows to specify in a graphical interface the main steps of the query optimization previously defined. This query rewriting is performed intuitively via the GUI by a decision maker. The process of implementing a pagelet is divided into two phases described below.

3.1 Designing Materialized Continuous Queries

First, the manager has to define all analyses on data streams that he wants to use in his pagelet.

Let us consider the following query: "What is the revenue achieved by the company every day from 01/01/1992 to 01/01/1993, considering DB at the most recent state?". The screenshot (Fig.3) shows a part of Decision Insight's user interface used to create a new analysis corresponding to Fig. 2. (A) indicates which rhythm do we want to link to the attribute. In our case we choose a one-day rhythm as we want to know the company's total revenue per day. (B) indicates the aggregation operation used to generate the analysis. (C) indicates the time-range to consider for the aggregation. In our example we fix at the last day. Finally (D) represents data inputs used to compute the analysis, which is the attribute "extendedPrice" of the LineItem. For each of these attributes, Decision Insight creates attributes via an underlying *materialized continuous query* in a transparent way for the user.

3.2 Designing On-Demand Queries

In the second phase, the manager chooses the form and the content that will be displayed on the pagelet. Fig.4 is a screenshot of Decision Insight's user interface for implementing a pagelet. (A) indicates the type of graphical element the manager wants to display, namely a historical curve. (B) indicates the time range of information to display on the pagelet. According to the query, we choose to display the whole current month. (C) indicates the information to be displayed. Based on the provided information, Decision Insight creates a pagelet and an underlying *on-demand query* in charge of updating the pagelet content (Fig.2).

4. EXPERIMENTS

4.1 Bi-temporal DB Benchmarks

A bi-temporal benchmark can be used in our case since it offers bi-temporal data that can be used to simulate real-time data. To the best of our knowledge, the TPC-BiH [17] is the most complete bi-temporal benchmark. TPC-BiH is an extension of TPC-H [11] and measures the performances of a DBMS used by a decision support system. The benchmark also includes a data generator producing a workload based on 9 categories of business transactions (New Order, Cancel Order, Update Stock, etc).

4.2 Database Populating

From the initial data produced by the TPC-BiH data generator, we generate a stream of events $< id, data, T >$. Each event corresponds to an updating instruction addressed to the database. id is the event type, e.g "insert a new order" or "insert a new customer". $data$ is the information handled by the event and T is the timestamp when the event occurred. The events are ordered according to the attribute T, so we can simulate a real-time workload. The initial TPC-BiH dataset has a size of 400MB. The generated data stream contains 3620761 events (Table 3).

We also introduce a scaling factor "*sf*" to fix the rate of the data stream. For the initial data stream $sf = 1$. All data streams with a

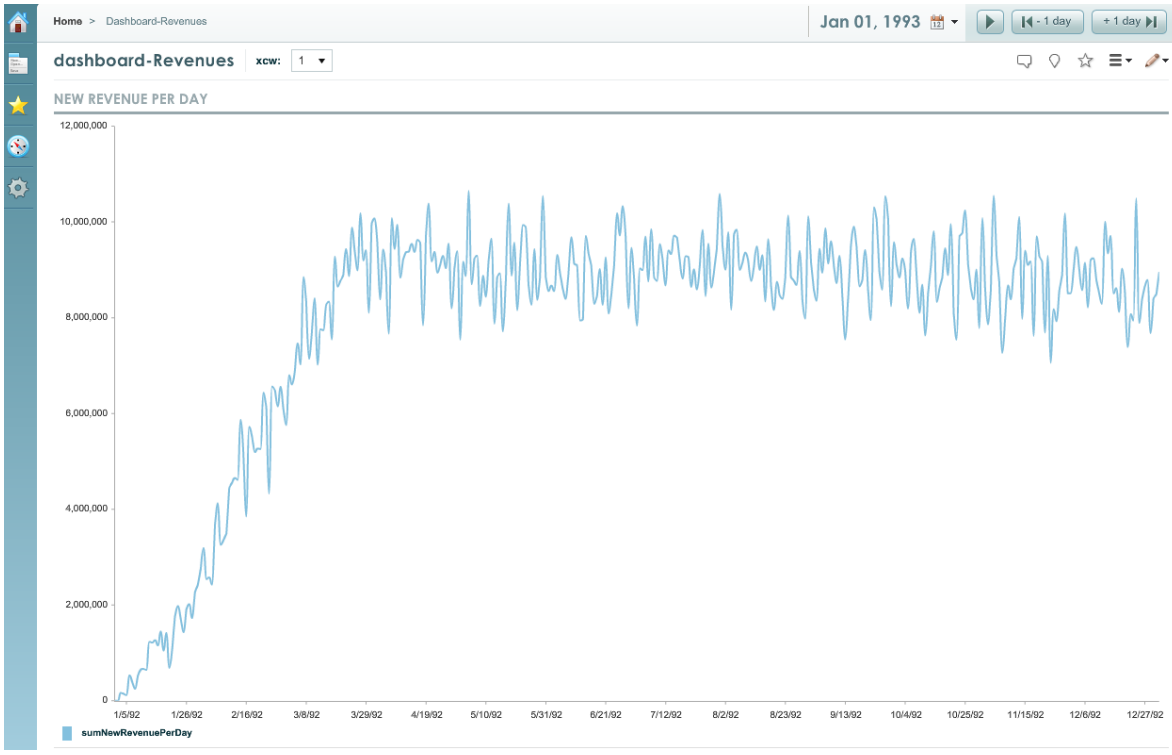

Figure 2: Snapshot of a pagelet displaying the daily evolution of revenues

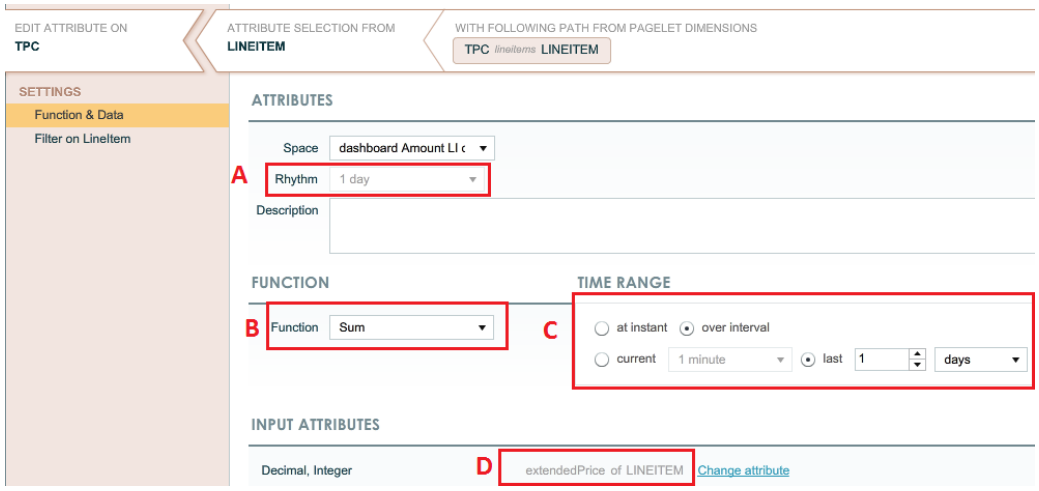

Figure 3: Decision Insight's interface to implement a continuous query

higher sf are generated by duplicating sf times each of its events.

4.3 Queries

We have implemented two examples of typical queries used in BAM. They are simple so that they can be easily expressed. Those queries are frequently executed by a GUI which requires rapid response times.

4.3.1 Query 1

This first query, (listing 11), aims at answering the following business question: "What is the sum of new revenues for the company every day from 1/1/1992 to now considering the most recent data?". We redefine this query as one *materialized continuous query* "Q1-Cont" (Listing 12) and one *on-demand query* "Q1-OnD" (Listing 13).

Listing 11: Q1: New Revenue per day

```
SELECT Ryhthm_ld.vtb as vtb, Ryhthm_ld.vte as vte,
```

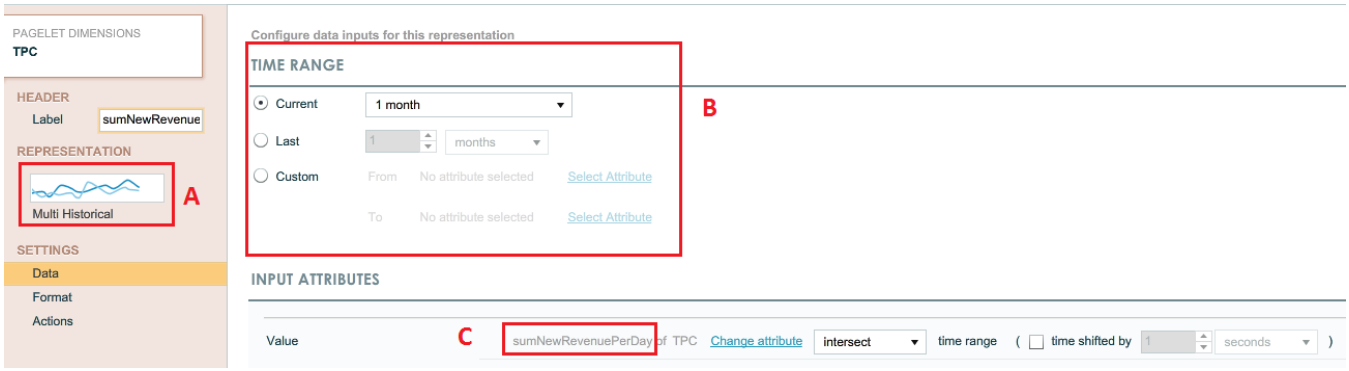

Figure 4: Decision Insight's interface for implementing an On-Demand query

Table 3: Number of operations per table

Relation	# of insertions	# of updates	# of deletions
Region	5	0	0
Nation	25	0	0
Supplier	1000	0	0
Part	20000	49861	0
Customer	164668	253430	0
Partsupp	80000	352391	0
Orders	348026	681103	8452
LineItems	939670	699310	22820

```
  SUM(extendedPrice) as totalRevenuePerDay
FROM LineItems, Ryhthm_1d
WHERE
  '01/01/1992'=<Ryhthm_1d.vtb AND
  Ryhthm_1d.vtb < '01/01/[YEAR]' AND
  Ryhthm_1d.vtb <= LineItems.vtb AND
  LineItems.vtb< Ryhthm_1d.vte
GROUP BY Ryhthm_1d.vtb;
```

Listing 12: Q1-Cont: New Revenue per day

```
SELECT SUM(extendedPrice) as totalRevenuePerDay,
  [vtInterval].begin as vtb, [vtInterval].end as vte
FROM LineItems
WHERE [vtInterval].begin <= vtb AND
  vtb < [vtInterval].end;
-- [vtInterval]: a rhythm interval
```

Listing 13: Q1-OnD: New Revenue per day

```
SELECT vtb, vte, aggr as newRevenuePerDay
FROM totalRevenuePerDay
WHERE '01/01/1992' <= vtb AND
  vtb < '01/01/[YEAR]';
```

4.3.2 *Query 2*

The query given in the (listing 14) aims at answering the following business question: "What is the number of orders per status for every day from 1/1/1990 to now considering the most recent data?". We redefine this query as one *materialized continuous query* "Q2-Cont" (Listing 15) and one *on-demand query* "Q2-OnD" (Listing 16).

Listing 14: Q2: Number orders per status and per day

```
SELECT Ryhthm_1d.vtb as vtb, Ryhthm_1d.vte as vte,
  SUM(extendedPrice) as totalRevenuePerDay
FROM Orders, Ryhthm_1d
WHERE
  '01/01/1992'=<Ryhthm_1d.vtb AND
  Ryhthm_1d.vtb < '01/01/[YEAR]' AND
  Ryhthm_1d.vtb <= order.vtb AND
  order.vtb< Ryhthm_1d.vte;
GROUP BY Ryhthm_1d.vtb, orderid;
```

Listing 15: Q2-Cont: Number of orders per status and per day

```
SELECT COUNT(*) as numberOrdersPerDay,
  [vtInterval].begin as vtb, [vtInterval].end vte
FROM Orders
WHERE [vtInterval].begin <= vtb AND
  vtb < [vtInterval].end
GROUP BY orderid;
```

Listing 16: Q2-OnD: Number of orders per status and per day

```
SELECT vtb, vte, numberOrdersPerDay
FROM Orderstatus
WHERE '01/01/1992' <= vtb AND
  vtb <'01/01/[YEAR]';
```

4.4 Experimental Results

In this section we present the results of experiments conducted to assess the performances of our approach. To do this, we compare system performances with and without using our optimization. We also present some measures of the overhead induced by our optimization. Experiments were executed on a physical machine which runs an Ubuntu 10.04, equipped with 12GB of RAM, an Intel i7 processor with 8 cores at 2.8GHz and a 4TB of RAID storage.

4.4.1 *Response Time*

In this test, we point out the interest of our approach in reducing Decision Insight's time response. We run two experiments: in the first, we evaluate the impact of the time range size on the execution time of Q1 and Q2 ($sf = 1$) while in the second, we vary the value of sf.

Fixed Scalar Factor

We inject stream concerning the period [1/1/1992, 1/1/1999[. At the beginning of each new year of the simulation period, we execute once Q1 and Q2 using a new value of the parameter "[YEAR]". We compare two versions of each query: the optimized version, using our approach based on continuous queries (Q1-OnD and Q2-OnD), and a classical version, where the result is computed whenever the

query arrives (Q1 and Q2). We collect the execution times of these queries and represent them on Fig.5.

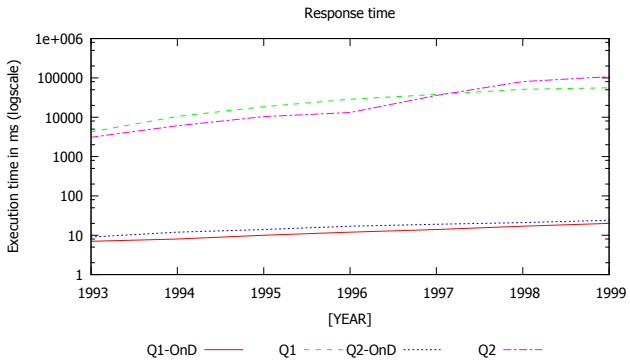

Figure 5: Query response time while varying window size

We can notice that optimized versions of queries outperform the rest by at least a factor of 100. For each day of the query interval, Q1-OnD accesses one value which is the materialized result of the underlying continuous query. Q1, however, accesses the original data, i.e about 200 items for each day.

Varying Scalar Factor
In this experiment, we assess our approach when we vary the data stream rate. The experimental conditions are similar to the previous test. We vary the value of sf from 1 to 6. For each value of sf, we inject the stream that concerns the period [1/1/1992, 1/1/1995[. Following injection, queries Q1 and Q2 are successively executed with and without optimization. The queries are executed with [YEAR]=1995.

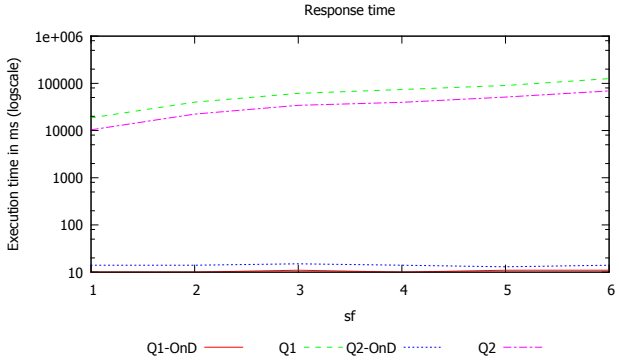

Figure 6: Query response time while varying data stream rate

When the data stream rate increases, the query execution time of the non-optimized queries increases, else it remains stable.

4.4.2 Precomputation Overhead

Previous tests demonstrate the advantage of our approach in reducing the response time of the system. However it induces a CPU and disk storage overhead.

Fixed Scalar Factor
The experimental conditions are similar to the test for response time/fixed scalar factor, except that we use only Q1. For each day of the simulated period, we collect the CPU time of Q1-Cont. We also collect the CPU time to execute Q1-OnD and Q1. Fig.7 shows the results of the experiment: one curve represents the CPU consumption of Q1, while the other is the sum of the CPU consumption of Q1-cont and Q1-OnD.

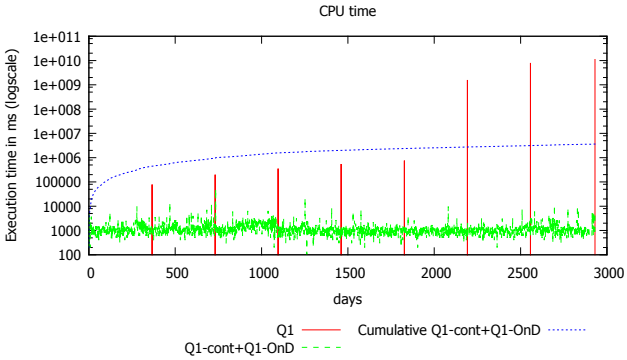

Figure 7: Continuous query computation overhead

It appears that the optimized approach requires a CPU overhead throughout the simulation time. However, it smooths the CPU consumption curve and avoids peaks at query time and thus system overload. We also notice that as from the 2000th day of simulation, the CPU Q1 cost is at least 100 times greater than the CPU required to compute Q1-cont and Q1-OnD. This means that for a query using a large time interval (6 years), the overhead induced by our approach has no impact on query processing performance.

Varying Scalar Factor
In this experiment we assess the cost of our approach as we vary the stream rate using the parameter sf. For each stream, we first inject data stream corresponding to the period [1/1/1992, 1/1/1995[, then we execute Q1 with [YEAR]= 1995. We collect the CPU time to perform Q1 and Q1-OnD. We also collect the average CPU time of Q1-cont per day and the total sum of all CPU time consumption of Q1-cont during the simulation. The results are represented in (Fig.8).

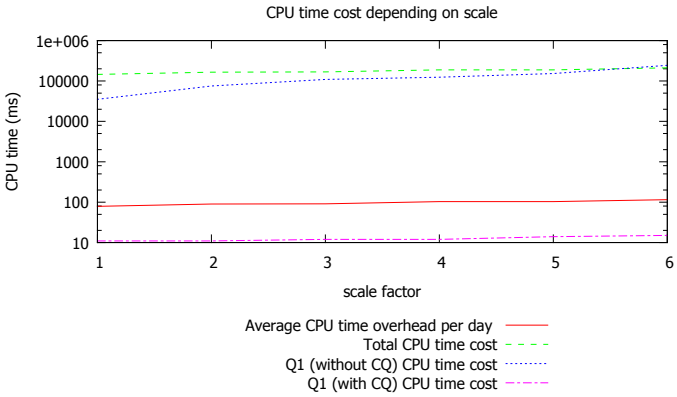

Figure 8: CPU time as a function of scale factor

Whenever $sf >= 6$, our approach does not have any CPU overhead.

4.4.3 Concurrent Query Execution

In this test, we simulate several users interacting with the system. We have performed two experiments: one where we vary the number of concurrent queries and another where we vary sf.

Fixed Scalar Factor

In this experiment, we use a dataset where $sf = 1$. We first populate the system with data corresponding to the period [1/1/1992, 1/1/1999[. Following system population, we execute concurrently several instances of the query Q1 with [YEAR]=1999. Then we get the CPU time required to execute them all. Fig.9 shows the results of this experiment where we varied the number of simultaneous executed queries from 1 to 20.

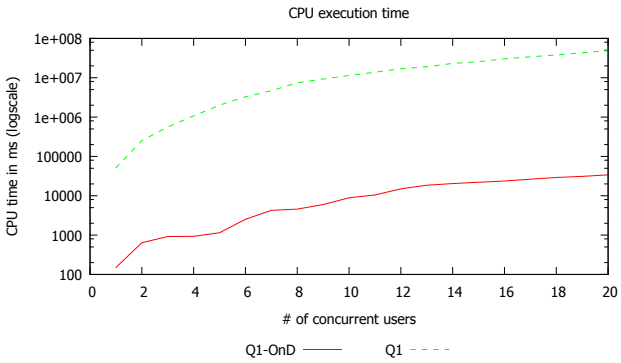

Figure 9: Concurrent query execution

As shown in Fig.9, our approach is quite adapted for execution of concurrent queries.

Varying Scalar Factor

In this experiment, we explore the impact of the data stream throughput on the execution of concurrent queries. We first populate the system with data that corresponds to the period [1/1/1992, 1/1/1995[. Then we execute 10 concurrent queries, corresponding to 10 users.

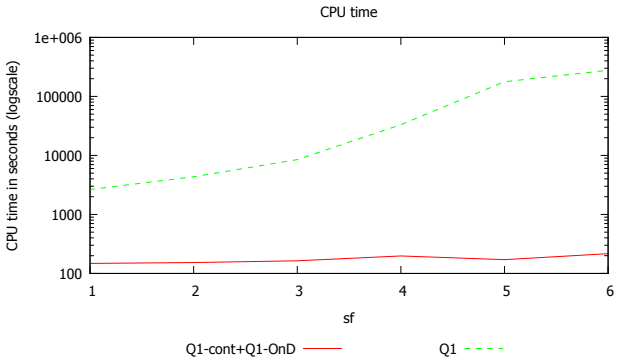

Figure 10: Concurrent query execution while varying sf

As in the previous test, we observe the advantage of the proposed optimization due to the increasing number of queries accessing the results of continuous queries as historical data.

5. RELATED WORKS

To the best of our knowledge, Chandrasekaran and Franklin were the first to address the topic of combining real-time data with historical data [9] in the academic field. They noted that the main performance issue for those systems was the I/O cost induced by gathering historical data, which decreases drastically live data stream processing performance. They proposed a framework using some data reduction techniques for historical data to limit I/O cost (see also [5]). Their framework data reduction level to be adapted with respect to current available resources. They defined three approaches to perform data reduction techniques: *OnWriteReplicate*, *OnRead-Modify* and *Hybdrid approach*. The first approach is based on the fact that random disk I/Os are expensive. Data reduction is performed continuously as soon as data is collected by the system. Thus at query time, the global query can access pre-computed results when needed. Nevertheless, pre-computed results can never be accessed by global queries. The second approach consists in performing the data reduction at query time only. The price to be paid can be very high for delivering timely information. The third approach combines the two previous approaches and shares the work between data arrival and query time. In this approach there is a single copy of the stream stored on disk and divided into separate batches. Each batch is divided into a fixed number of blocks. Tuples are randomly inserted in different blocks of the current batch. Once one block is filled, the entire run is flushed on disk. At query time, the system only accesses a fraction of blocks of runs according to a sampling rate.

With respect to our contribution, we use the OnWriteReplicate approach which turns out to be effective for BAM applications. Postponing query processing when the system is at a lower load is not a new idea, see for instance [10] in a BI context. Load shedding techniques have been investigated in DSMS to come up with high data throughput [32]. Such techniques are quite different from the proposition made in this paper. The application context of Business Activity Monitoring does not fit perfectly into either of those two main areas, i.e. neither the data volume is expected to be as large as in data-warehouse applications, nor the data throughput is expected to cause the system to collapse as in some DSMS applications. BAM applications lie somewhere between these two kinds of applications.

The topic of materialized views has been widely discussed in DBMS in general [21, 25] and in real-time DBMS in particular [2], since materialized views are often considered to reduce query execution time. These works mainly suppose the case of *snapshot views*, i.e views that maintain only the last state of data. This assumption is not sufficient in our case since we need to store its whole history. There were some works concerning temporal materialized views [34, 14, 13]. Yet the do not address the real-time case.

The concept of *rhythm* is close to the concept of *Granularity* that has already been defined in litterature [6] and has been implemented in some products. We can quote Teradata' DBMS that extends SQL with the key word "EXPAND ON". There is also Kx System's product KDB+ whose processing language supports such feature.

Many commercial products addressing business monitoring exist, among which we quote Kx System [4], spunk and main DBMS players like DB2, Oracle and Teradata. However, as far as we know, no one covers all features offered by Decision Insight. KDB+, for example, handles both real-time and historical data and is based on a column-store database dedicated to handle time-series data. Yet, it only supports valid time dimension.

6. CONCLUSION

[4] http://kx.com/

In this paper, we have introduced Decision Insight, a comprehensive data-intensive decision support system that combines both Business Activity Monitoring (BAM) and business intelligence capabilities.
Given a bi-temporal query on historical and live data, the optimization technique of Decision Insight is based on a decomposition of the initial query into several continuous queries plus one on-demand query, which accesses the materialized data of previous queries. This technique has been implemented in Decision Insight, and experiments have been conducted on the TPC-BiH benchmark (variant of the TPC-H benchmark). Results show that Decision Insight is able to deliver very fast responses to decision-makers, which is a very strong requirement in BAM applications.

In future works, we aim to study multi-query optimization techniques to share as much as possible the processing of a set of complex bi-temporal queries [26, 8].

7. REFERENCES

[1] D. J. Abadi, S. R. Madden, and N. Hachem. Column-stores vs. row-stores: how different are they really? In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 967–980. ACM, 2008.

[2] B. Adelberg, H. Garcia-Molina, and B. Kao. Applying update streams in a soft real-time database system. In *ACM SIGMOD Record*, volume 24, pages 245–256. ACM, 1995.

[3] A. Ait Ouassarah, N. Averseng, X. Fournet, J.-M. Petit, R. Revol, and V.-M. Scuturici. Understanding Business Trends from Data Evolution with Tornado (demo). In *Int. Conf. on Data Engineering (ICDE 2015)*. IEEE, May 2015.

[4] A. Arasu, S. Babu, and J. Widom. An abstract semantics and concrete language for continuous queries over streams and relations. 2002.

[5] D. Barbar'a, W. DuMouchel, C. Faloutsos, P. J. Haas, J. M. Hellerstein, Y. Ioannidis, H. Jagadish, T. Johnson, R. Ng, V. Poosala, et al. The new jersey data reduction report. In *IEEE Data Engineering Bulletin*. Citeseer, 1997.

[6] C. Bettini, C. E. Dyreson, W. S. Evans, R. T. Snodgrass, and X. S. Wang. A glossary of time granularity concepts. In *Temporal databases: Research and practice*, pages 406–413. Springer, 1998.

[7] P. A. Boncz, M. Zukowski, and N. Nes. Monetdb/x100: Hyper-pipelining query execution. In *CIDR*, volume 5, pages 225–237, 2005.

[8] G. Candea, N. Polyzotis, and R. Vingralek. A scalable, predictable join operator for highly concurrent data warehouses. *Proceedings of the VLDB Endowment*, 2(1):277–288, 2009.

[9] S. Chandrasekaran and M. Franklin. Remembrance of streams past: Overload-sensitive management of archived streams. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 348–359. VLDB Endowment, 2004.

[10] S. Chaudhuri and U. Dayal. An overview of data warehousing and olap technology. *ACM Sigmod record*, 26(1):65–74, 1997.

[11] T. P. P. Council. Tpc-h benchmark specification. *Published at http://www.tcp.org/hspec.html*, 2008.

[12] C. Dyreson, F. Grandi, W. Käfer, N. Kline, N. Lorentzos, Y. Mitsopoulos, A. Montanari, D. Nonen, E. Peressi, B. Pernici, J. F. Roddick, N. L. Sarda, M. R. Scalas, A. Segev, R. T. Snodgrass, M. D. Soo, A. Tansel, P. Tiberio, and G. Wiederhold. A consensus glossary of temporal database concepts. *SIGMOD Rec.*, 23(1):52–64, Mar. 1994.

[13] H. V. Jagadish, I. S. Mumick, and A. Silberschatz. View maintenance issues for the chronicle data model. In *Proceedings of the fourteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 113–124. ACM, 1995.

[14] C. S. Jensen, L. Mark, and N. Roussopoulos. Incremental implementation model for relational databases with transaction time. *Knowledge and Data Engineering, IEEE Transactions on*, 3(4):461–473, 1991.

[15] C. S. Jensen and R. Snodgrass. Temporal specialization and generalization. *Knowledge and Data Engineering, IEEE Transactions on*, 6(6):954–974, 1994.

[16] C. S. Jensen and R. T. Snodgrass. Temporally enhanced database design., 2000.

[17] M. Kaufmann, P. M. Fischer, N. May, A. Tonder, and D. Kossmann. Tpc-bih: A benchmark for bitemporal databases, 2013.

[18] M. Kaufmann, P. Vagenas, P. M. Fischer, D. Kossmann, and F. Färber. Comprehensive and interactive temporal query processing with sap hana. *Proceedings of the VLDB Endowment*, 6(12):1210–1213, 2013.

[19] J. Krämer and B. Seeger. *A temporal foundation for continuous queries over data streams*. Univ., 2004.

[20] K. Kulkarni and J.-E. Michels. Temporal features in sql: 2011. *ACM SIGMOD Record*, 41(3):34–43, 2012.

[21] P.-Å. Larson and H. Yang. *Computing queries from derived relations: Theoretical foundation*. Citeseer, 1987.

[22] D. Luckham. *The power of events*, volume 204. Addison-Wesley Reading, 2002.

[23] D. W. McCoy. Business activity monitoring: Calm before the storm. *Gartner Research*, 2002.

[24] F. Reiss, K. Stockinger, K. Wu, A. Shoshani, and J. M. Hellerstein. Enabling real-time querying of live and historical stream data. In *Scientific and Statistical Database Management, 2007. SSBDM'07. 19th International Conference on*, pages 28–28. IEEE, 2007.

[25] H. A. Schmid and P. A. Bernstein. A multi-level architecture for relational data base systems. In *Proceedings of the 1st International Conference on Very Large Data Bases*, pages 202–226. ACM, 1975.

[26] T. K. Sellis. Multiple-query optimization. *ACM Transactions on Database Systems (TODS)*, 13(1):23–52, 1988.

[27] R. Snodgrass. The temporal query language tquel. *ACM Transactions on Database Systems (TODS)*, 12(2):247–298, 1987.

[28] R. Snodgrass, M. Böhlen, C. Jensen, and A. Steiner. Adding valid time to sql/temporal. ansi x3h2-96-151r1, iso–ansi sql/temporal change proposal, iso. Technical report, IEC JTC1/SC21/WG3 DBL MCI-142, 1996.

[29] R. T. Snodgrass, M. H. Böhlen, C. S. Jensen, and A. Steiner. Adding transaction time to sql/temporal. *ISO-ANSI SQL/Temporal Change Proposal, ANSI X3H2-96-152r ISO/IEC JTC1/SC21/WG3 DBL*, 1101:143, 1996.

[30] R. T. Snodgrass, J. Gray, and J. Melton. *Developing time-oriented database applications in SQL*, volume 42. Morgan Kaufmann Publishers San Francisco, 2000.

[31] M. Stonebraker, D. J. Abadi, A. Batkin, X. Chen, M. Cherniack, M. Ferreira, E. Lau, A. Lin, S. Madden, E. O'Neil, et al. C-store: a column-oriented dbms. In *Proceedings of the 31st international conference on Very*

large data bases, pages 553–564. VLDB Endowment, 2005.

[32] N. Tatbul, U. Çetintemel, S. Zdonik, M. Cherniack, and M. Stonebraker. Load shedding in a data stream manager. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 309–320. VLDB Endowment, 2003.

[33] H. J. Watson and B. H. Wixom. The current state of business intelligence. *Computer*, 40(9):96–99, 2007.

[34] J. Yang and J. Widom. Maintaining temporal views over non-temporal information sources for data warehousing. In *Advances in Database TechnologyÑEDBT'98*, pages 389–403. Springer, 1998.

ISSN 2429-4586

An approach for measuring RDF data completeness

Fayçal Hamdi
Cedric Lab, Conservatoire National des Arts et Métiers (CNAM)
Paris, France
faycal.hamdi@cnam.fr

Samira Si-Said Cherfi
Cedric Lab, Conservatoire National des Arts et Métiers (CNAM)
Paris, France
samira.cherfi@cnam.fr

ABSTRACT

With the increasing use of the Web of Data and the development of data based analysis applications, there is a real need for developing suitable methods and techniques for evaluating and help ensuring web data quality. Among Quality dimensions, completeness is recognized as difficult to evaluate, as it often relies on gold standards and/or a reference data schema that are neither always available nor realistic from a practical point of view. In this paper, we propose an approach for Linked Data completeness assessment. The approach is a two-step process: first, mining from a data source, a schema that reflects the actual representation of data, which, in the second step, is used for completeness evaluation. The paper presents both theoretical background and experimental results performed on real-world RDF linked datasets.

Keywords

Linked Data, RDF Data Quality, Completeness, Quality Evaluation

1. INTRODUCTION

Nowadays, with the development of web technologies, data is considered as a strategic asset for increasing companies' revenue. This phenomenon is enhanced thanks to the crowd sourced community efforts that led to the availability of huge structured data sets covering a wide variety from several sources such as civil society, local communities and domain experts. This data is being released continuously and, thanks to its digital format, it is mainly suitable for computerized usage leading to the development of new techniques, technologies, practices and methodologies aiming to better understand business and market needs. As a consequence, this data, accessible from a wide range of users, empowers and even influences decision making processes. According to a report by the McKinsey Global Institute [19], by using social technologies, companies could increase margins by as much as 60 percent. In research area, the exploration of semantic links between datasets will enable novel analyzes to be performed. For example, linked open drug data aims to connect previously unlinked results from clinical trials, gene expression assays, and chemical testing [27].

However, as data publishing does not require special expert knowledge or skills, availability of data does not always guarantees its usability. This means that the quality of published data is not as good as we could expect leading to a low added value and low reliability of the derived conclusions. Information quality has been extensively studied in the context of relational databases [29, 21]. However, in the context of Web of Data, this is an emerging issue and a real challenge for future research. Indeed, Linked Open Data[1] (LOD) through HTTP URI's offer an infrastructure for publishing and navigating through data without ensuring the quality of its usage [3]. As a consequence, there is a real need for developing suitable methods and techniques for evaluating and help ensuring web data quality.

Quality of the Web of Data attracted recent interest and main contributions go into two directions. The first one relies on user contribution to qualify the quality of data and is thus subjective. The second one, more objective, mainly concentrates on data provenance. In this paper we are more precisely concerned with data completeness as a dimension of data quality and the way to assess it. Indeed, data completeness is recognized as an important quality dimension and providing completeness information about a data source helps increasing the confidence of such a source.

Data completeness has two facets. The first one analyzes whether all data is available. Such completeness is known as structural completeness [1] and requires a reference benchmark or gold data set as completeness reference. This vision is more related to the amount of data provided and its relevance to answer users requests. The second facet entails having a value for each property of the data. This latter is then said complete if all necessary values are recorded. A traditional way to measure completeness in this case is the rate of missing values. This subsumes that data rely on an agreed and well-designed schema in which properties are equally relevant. Indeed, assessing completeness as the rate of missing values does not take into account the fact that missing a marginal property should have a less importance than missing an essential value. In the context of web data

[1] `http://lod-cloud.net`

(c) 2015, Copyright is with the authors. Published in the Proceedings of the BDA 2015 Conference (September 29-October 2, 2015, Ile de Porquerolles, France). Distribution of this paper is permitted under the terms of the Creative Commons license CC-by-nc-nd 4.0.
(c) 2015, Droits restant aux auteurs. Publié dans les actes de la conférence BDA 2015 (29 Septembre-02 Octobre 2015, Ile de Porquerolles, France). Redistribution de cet article autorisée selon les termes de la licence Creative Commons CC-by-nc-nd 4.0.
BDA 2015, 29 septembre au 02 octobre 2015, Ile de Porquerolles, France.

these two assumptions are not satisfied.

Let's consider, for instance, a collaboratively built dataset. In this case, the traditional top down vision of a predefined schema is no true. Both data and underlying schema evolve continuously as data are described by several communities having different views and needs. The challenge is thus to provide a suitable completeness assessment method taking into account the absence of an agreed schema. Accordingly, we adopt in the remaining of the paper the definition from [28] for completeness as the ability to represent every meaningful state of the represented real world system. This definition highlights two interesting aspects. The first one is the notion of state considering the combination of properties rather than their individual values. The second is the notion of meaningful state' suggesting that completeness should take into account the relevance of properties when analyzing missing values.

We aim to provide an approach for RDF data completeness assessment. Our approach could be applied based on a reference schema that is inferred from data values. This schema will represent the meaningful state of the considered dataset. In summary, we make the following contributions:

1. We use a mining approach to infer a schema from data, as we consider that no predefined schema exists.

2. We introduce a novel approach for data completeness assessment based on both the inferred schema and the data.

3. We experimentally assess our approach by using real-world datasets as a case study. We analyzed the relationship between the completeness and the number of properties in the inferred schema, and the completeness and the robustness of the calculation when dataset size varies.

The remainder of this paper is organized as follows: section 2 presents a motivating example introducing the problem tackled in this article; section 3 gives a formal description of the problem; section 4 details the mining-based approach; section 5 presents and analyzes a set of experiments; section 6 summarizes a related literature on the subject while section 7 draws conclusions and future research directions.

2. MOTIVATING EXAMPLE

We illustrate in this section the main idea behind our approach through an example that shows the issues and the difficulties encountered in the calculation of a dataset completeness. Let consider the set of scientists described in the well-known open linked dataset, DBpedia. We would like to know, when querying this dataset about a particular scientist, if the information provided for this scientist are complete or not. We would like also to know if all the scientists in this dataset are well described.

To do so, the first intuition consists of comparing the properties used in the description of each scientist with a reference scientist schema (ontology). For example, in DBpedia, the class *Scientist*[2] has a list of 4 properties (e.g. *doctoralAdvisor*), but these properties are not the only ones used in the description of a scientist (e.g. the *birthdate* property is not present in this list). Indeed, the class *Scientist* has a super class called *Person*. So, the description of a scientist may also take into account the properties of this class. Therefore, to obtain an exhaustive list of the whole properties used in the description of a scientist, we have to calculate the union of the set of properties of the class *Scientist* and all its ancestors. For our example, the reference scientist schema that we called *Scientist_Schema* could be calculated as follows:

$$Scientist_Schema = \{Properties\ on\ Scientist\} \cup \{Properties\ on\ Person\} \cup \{Properties\ on\ Agent\} \cup \{Properties\ on\ Thing\}$$

$$\text{such that: } Scientist \sqsubseteq Person \sqsubseteq Agent \sqsubseteq Thing$$

Thus, the completeness of a scientist description (e.g. *Albert_Einstein*) will be the proportion of properties used in the description of this scientist to the total number of properties in *Scientist_Schema*. In the case of DBpedia, with a simple SPARQL query[3], we can obtain the size of *Scientist_Schema*, which is equal to 664 (A-Box properties). So, the completeness of the description of *Albert_Einstein* could be calculated as follows:

$$Comp(Albert_Einstein) = \frac{|Properties\ on\ Albert_Einstein|}{|Scientist_Schema|} = \frac{21}{664} = 4,21\%$$

However, on the one hand, we know that the description of a scientist does not possibly need all the properties of the schema. For example, the property *weapon* is far from being relevant in the description of *Albert_Einstein* (in DBpedia, this property is not used in any scientist instances, whether it is part of the Scientist_Schema) and cannot have the same importance as, for instance, the *name* or the *university* of a scientist. This is also the case of several properties in *Scientist_Schema*. In addition, a description, with 21 A-Box properties[4] (and 66 T-Box and other external properties), of a scientist seems to be actually a good representation and provides enough useful information about this scientist. Thus, considering that the description of *Albert_Einstein* has solely 4.21% as a value of completeness is quite reductive. On the other hand, when we calculate, for example, the completeness of the first 1000 scientists, we obtained a value of completeness equals to 1.37% (lower that the one obtained for the so "famous" scientist *Albert_Einstein*). This is due to the fact that several "not famous" scientist has a very few number of A-Box properties. Therefore, the completeness of the whole dataset gets lower regarding to the number of the "not famous" scientists (it certainly get even lower when it calculated for the 18,233 DBpedia scientists[5]).

[2] `http://mappings.dbpedia.org/server/ontology/classes/`
[3] Performed on: `http://dbpedia.org/sparql`
[4] `http://dbpedia.org/resource/Albert_Einstein`
[5] `http://wiki.dbpedia.org/Datasets/DatasetStatistics` (statistics of the DBpedia 2014 version for the english language)

We can finally conclude that, the completeness as calculated here, does not provide us with the relevant value regarding the real representation of scientists in the DBpedia dataset. Hence, to overcome this issue, there is a need to explore instances to get an idea about how they are actually describing and which properties, with the importance of each one, are used. Based on data mining, the approach that we propose in this paper, deals with this issue by extracting, from a set of instances (of the same class), the pattern of the most representative properties and calculates a completeness in respect to this pattern.

3. PROBLEM STATEMENT

The problem of completeness evaluation is not to find an absolute value but to compute the more suitable one for the context of use. In our situation, the context is a dataset representing a category or a set of categories such as *Actor* or *Organization*. Table 1 illustrates some instances of the *Actor* category in form of triples, taken from DBpedia.

Table 1: A sample of triples from DBpedia

Subject	Predicate	Object
Ben_Affleck	birthDate	1972-01-01
Ben_Affleck	residence	Los Angeles
Angelina_Jolie	birthDate	1975-06-04
Angelina_Jolie	citizenship	United_States
Adam_West	birthDate	1928-09-19
Adam_West	citizenship	American
Adam_West	residence	Ketchum,_Idaho

Each Category is described by a set of properties (predicates) and an instance of this category could have a value for all the properties or only for a subset of these properties. This subset is called transaction. Table 2 represents the set of transactions constructed from the triples of the table 1.

Table 2: Transactions created from triples

Instance	Transaction
Ben_Affleck	birthDate, residence
Angelina_Jolie	birthDate, citizenship
Adam_West	birthDate, citizenship, residence

More formally, lets define a dataset $\mathcal{D}$ to be triple (C, I_C, P), where C is the set of categories (e.g., *Actor*, *City*), I_C is the set of instances for categories in C (e.g., *Ben_Affleck* is an instance of the *Actor* category), and $P = \{p_1, p_2, ..., p_n\}$ is the set of properties (e.g $residence(Person, Place)$).

Let $\mathcal{T} = \{t_1, t_2, ..., t_m\}$ be a set of transactions with $\forall k, 1 \leq k \leq m : t_k \subseteq P$ be a vector of transactions over P, and $E(t_k)$ be the set of items in transaction t_k. Each transaction is a set of properties used in the description of the instances of the subset $\mathcal{I}' = \{i_1, i_2, ..., i_m\}$ with $\mathcal{I}' \subseteq I_C$ (e.g., properties used to describe the *Ben_Affleck* instance are: *birthDate* and *residence*). We consider $\mathcal{CP}$ the completeness of $\mathcal{I}'$ against properties used in the description of each of its instances.

Problem definition Given a dataset $\mathcal{D}$, a subset of instances $\mathcal{I}'$, and a set of transactions $\mathcal{T}$ constructed from $\mathcal{I}'$, we would like to calculate the completeness $\mathcal{CP}$ of $\mathcal{I}'$.

4. THE MINING-BASED APPROACH

Completeness at the data level assesses missing values [26]. This vision requires a schema that needs to be inferred from the data source. However, as we mentioned in the motivating example, considering a schema as the union of all properties used for a category description is not relevant. Indeed, this vision neglects the fact that missing values could express inapplicability [6]. A missing value is said inapplicable when it represents a property that is inapplicable to the particular object or instance. In this case, we could conclude that all properties are not equally important.

To take this aspect into consideration, we propose an approach that calculates the completeness of an input dataset by posing the problem as an itemset mining problem. Therefore, the process of inferring a reference schema will consider the real data, actually contained in the dataset. Our mining-based approach includes two steps:

1. **Properties mining**: Given a dataset $\mathcal{D}$, we first represent the properties, used for the description of the $\mathcal{D}$ instances, as a transactions vector. We then apply the well known FP-growth algorithm [13, 14] for mining frequent itemsets (we chose FP-growth for efficiency reasons. Any other itemset mining algorithm could, obviously, be used). Only a subset of these frequent itemsets, called "Maximal" [20, 11, 12], is captured. This choice is motivated by the fact that, on the one hand, we are interested in the *expression* of the frequent pattern and, on the other hand, the number of frequent patterns could be exponential when the transaction vector is very large (see Section 4.1 for details).

2. **Completeness calculation**: Once the set of maximal frequent itemsets $\mathcal{MFP}$ is generated, we use the apparition frequency of items (properties) in $\mathcal{MFP}$, to give each of them a weight that reflects how important the property is considered for the description of instances. Weights are then exploited to calculate the completeness of each transaction (regarding the presence or absence of properties) and, hence, the completeness of the whole dataset.

In the following we give a detailed description of each step.

4.1 Properties mining

Let $\mathcal{D}(C, I_C, P)$ be a dataset and $\mathcal{I}'$ be a subset of instances with $\mathcal{I}' \subseteq I_C$. We first initialize $\mathcal{T} = \phi$, $\mathcal{MFP} = \phi$. For each $i \in \mathcal{I}'$ we generate a transaction t. Indeed, each instance i is related to values (either resources or literals) through a set of properties. Therefore, a transaction t_k of an instance i_k is a set of properties such that $t_k \subseteq P$. Transactions generated for all the instances of $\mathcal{I}'$ are then added to the $\mathcal{T}$ set.

Example 1. Taking table 1, let $\mathcal{I}'$ be a subset of instances such that: $\mathcal{I}' = \{$*Ben_Affleck, Angelina_Jolie, Adam_West*$\}$. The set of transaction $\mathcal{T}$ would be:

$$\mathcal{T} = \{\{birthDate, residence\}, \{birthDate, citizenship\}, \{birthDate, citizenship, residence\}\}$$

The objective is then to compute the set of frequent patterns $\mathcal{FP}$ from the transaction vector $\mathcal{T}$.

Definition 1. (*Pattern*) Let $\mathcal{T}$ be a set of transactions. A pattern $\hat{P}$ is a sequence of properties shared by one or several transactions t in $\mathcal{T}$.

For any pattern $\hat{P}$, let $E(\hat{P})$ be the corresponding set of items (constitutes, in our case, of properties), and $T(\hat{P}) = \{t \in \mathcal{T} \mid E(\hat{P}) \subseteq E(t)\}$ be the corresponding set of transactions. $E(\hat{P})$ designates the *expression* of $\hat{P}$, and $|T(\hat{P})|$ the *support* of $\hat{P}$. A pattern $\hat{P}$ is frequent if $\frac{1}{|\mathcal{T}|}|T(\hat{P})| \geq \xi$, where ξ is a user-specified threshold.

Example 2. Taking table 2, let $\hat{P} = \{birthDate, residence\}$ and $\xi = 60\%$. $\hat{P}$ is frequent as its relative support (66.7%) is greater than ξ.

To find all the frequent patterns $\mathcal{FP}$, we used, as we motivated above, the FP-growth itemsets mining algorithm. However, according to the size of the transactions vector, the FP-growth algorithm could generate a very large $\mathcal{FP}$ set. As our objective is to see how a transaction (a description of an instance) is *complete* against a set of properties, we focus on the pattern *expression* (in terms of items it contains) instead of its *support*.

For completeness calculation, we need to select a pattern to serve as reference schema. This pattern should present a right balance between frequency and expressiveness. In itemset mining, a concept, called "Maximal" frequent patterns, allow us finding this subset. Thus, to reduce $\mathcal{FP}$, we generate a subset containing only "Maximal" patterns.

Definition 2. ($\mathcal{MFP}$) Let $\hat{P}$ be a frequent pattern. $\hat{P}$ is maximal if none of its proper superset is frequent. We define the set of Maximal Frequent Patterns $\mathcal{MFP}$ as:

$$\mathcal{MFP} = \{\hat{P} \in \mathcal{FP} \mid \forall \hat{P}' \supsetneq \hat{P} : \frac{|T(\hat{P}')|}{|\mathcal{T}|} < \xi\}$$

Example 3. Taking table 2, let $\xi = 60\%$ and the set of frequent patterns $\mathcal{FP} = \{\{birthDate\},\{residence\},\{citizenship\}, \{birthDate,residence\}\{birthDate,citizenship\}\}$. The $\mathcal{MFP}$ set would be:

$$\mathcal{MFP} = \{\{birthDate,residence\}, \{birthDate,citizenship\}\}$$

4.2 Completeness calculation

In this step our goal is to find a unique pattern that allows us evaluating the completeness of a transaction regarding this pattern. However, the maximal frequent patterns $\mathcal{MFP}$, that we generated in the previous steps, contains often several candidate patterns. In this case, our strategy consists of reducing the $\mathcal{MFP}$ set to obtain only one pattern. However, all the properties from $\mathcal{MFP}$ are not equally important. We consider for that two aspects; the apparition frequency of an item (a property) in $\mathcal{MFP}$, and the support of each $\hat{P} \in \mathcal{FP}$ containing this item. The result will be a *weighted* frequent pattern that we denote $\hat{P}_w$. This pattern will be used then as a reference schema in the completeness calculation.

Definition 3. (*Weighted frequent pattern*) Let $\mathcal{MFP}$ be a set of maximal frequent pattern. A weighted frequent pattern $\hat{P}_w$ is a set of couples $< p, w(p) >$ composed of a property p and a weight w associated to this property. The weight w is calculated as follows:

$$\forall p \in \bigcup_{i=1}^{n} E(\hat{P}_i) : w(p) = \frac{1}{|\mathcal{MFP}|} \sum_{i=1}^{n} \delta(p,i).\frac{|T(\hat{P}_i)|}{|\mathcal{T}|} \quad (1)$$

such that: p a *singleton*, $\hat{P}_i \in \mathcal{MFP}$ and

$$\delta(p,i) = \begin{cases} 1 & \text{if } p \in E(\hat{P}_i) \\ 0 & \text{otherwise} \end{cases}$$

The equation 1 takes into account, as we mentioned above, the frequency of a property p by checking, via the δ parameter, its presence on the itemset of each maximum frequent pattern. As the weight is calculated relatively to the number of transactions in $\mathcal{T}$ (to get a relative *support*), its value will range between 0 and 1.

Example 4. Let $\mathcal{MFP} = \{\{birthDate,residence\}, \{birthDate,citizenship\}\}$ where both itemsets have a *support* of 67%. The set of the weighted frequent patterns $\hat{P}_w$ would be:

$$\hat{P}_w = \{\{birthDate, 0.67\}, \{residence, 0.33\}, \{citizenship, 0.33\}\}$$

Once we obtained the weighted pattern $\hat{P}_w$, we carry out for each transaction, a comparison between its corresponding properties and the weighted properties of $\hat{P}_w$. We get, therefore, an indication about the completeness of each transaction $t \in \mathcal{T}$.

Definition 4. (*Completeness* $\mathcal{C}$) Let $\mathcal{I}'$ a subset of instances, $\mathcal{T}$ the set of transactions constructed from $\mathcal{I}'$, and $\hat{P}_w$ the set of weighted frequent properties. The completeness of $\mathcal{I}'$ corresponds to the completeness of its transaction vector $\mathcal{T}$ obtained by calculating the average of the completenesses of each transaction regarding $\hat{P}_w$. Therefore, we define the completeness $\mathcal{CP}$ of a subset of instance $\mathcal{I}'$ as follows:

$$\mathcal{CP}(\mathcal{I}') = \frac{1}{|\mathcal{T}|} \sum_{k=1}^{|\mathcal{T}|} \sum_{j=1}^{n} \frac{w(p_j \cap E(t_k))}{\mathcal{W}} \quad (2)$$

such that:

$$w(\emptyset) = 0,\ \mathcal{W} = \sum_{j=1}^{n} w(p_j),\ p_j \in \bigcup_{j=1}^{n} E(\hat{P}_j), \text{ and } \hat{P}_j \in \mathcal{MFP}$$

It is worth noting that, the weights of the properties allow us to evaluate the completeness of a given transaction with respect to a context, which is in our case, the properties used in the all transactions. The Algorithm 1 shows the pseudo-codes for calculating $\mathcal{CP}(\mathcal{I}')$.

Example 5. Let $\xi = 60\%$. The completeness of the subset of instances in table 1 regarding $\hat{P}_w$ = {{birthDate,0.67}, {residence,0.33},{citizenship,0.33}}, would be:

$$\mathcal{CP}(\mathcal{I}') = (2*(0.33+0.67)+(0.33+0.33+0.67))/1.33*3 = 0.83$$

This value corresponds to the completeness average value for the whole dataset regarding the inferred schema $\hat{P}_w$.

Algorithm 1 Completeness calculation

Input: $\mathcal{D}, \mathcal{I}', \xi$
Output: $\mathcal{CP}(\mathcal{I}')$
  for each $i \in \mathcal{I}'$ **do**
    $t_i = \lvert p_1 \quad p_2 \quad \ldots \quad p_n \rvert$
    $\mathcal{T} = \mathcal{T} + t_i$
  ▷ *Properties mining*
  $\mathcal{MFP} = Maximal(FP\text{-}growth(\mathcal{T}, \xi))$
  $\hat{P}_w =< nil, nil >$
  for each $\hat{P} \in \mathcal{MFP}$ **do**
    for each $p \in \hat{P}$ **do**
      ▷ *Using equation 1*
      $w(p_i) = CalculateWeight(p, \mathcal{MFP}, \mathcal{T})$
      $\hat{P}_w.put(p_i, w(p_i))$
  ▷ *Using equation 2*
  return $CP(\mathcal{I}') = CalculateCompleteness(\mathcal{I}', \mathcal{T}, \hat{P}_w)$

5. EMPIRICAL EVALUATION

This section is devoted to a set of experiments aiming to evaluate our approach against a variation of its underlying parameters. The evaluation methodology explores the behavior of completeness metric regarding two sides. The first concerns the impact of the number of instances whereas the second is related to the user-specified threshold ξ. The experiments were performed on two well-known real-world datasets, publicly available on the Linked Open Data (LOD). The first one, DBpedia, is a large knowledge base composed of structured information extracted collaboratively from Wikipedia. It describes currently 4.58 million things. The second dataset, Freebase, is a large collaborative knowledge base of the world's information, built mainly from data provided by its community members. The last version of Freebase (before its migration to Wikidata) includes approximately 47.3 million topics and 2.9 billion facts.

5.1 Dataset description

For the evaluation of the robustness of our approach we are looking to investigate its behavior regardless the nature of data. To do so, for each dataset we have chosen a couple of categories from different natures. For DBpedia, we studied the completeness of instances that have as types the following categories: $C = \{$*Populated Place*, *Organisation*, *Actor*, *Athlete*$\}$, and for Freebase the instances that have as types the following ones (which are equivalent or close to those of DBpedia): $C = \{$*Citytown*, *Organization*, *Football Player*, *Museum*$\}$. Indeed, our objective is not to compute an absolute completeness of a source. In practice, the completeness of the whole source is of marginal interest since queries often concern a subset of data. We thus analyzed the behavior and robustness of our completeness measure for various categories with different sizes. This is motivated by the fact that good completeness values, for a desired category, remain attractive even if the overall completeness (average or other aggregate measure) or the completeness of categories that do not interest the user are not satisfactory. Besides, a completeness of 99% may be uninteresting if the schema is ridiculously small.

In the first step of our experiments, we performed queries (SPARQL for DBpedia and MQL for Freebase) on datasets endpoints to extract data concerning each category. We then constructed the set of corresponding transactions $\mathcal{T}$. A transaction vector is constituted of sequences of properties deduced from instances belonging to a single category (e.g. the set of *Actors* in DBpedia). The set of transactions[6] is then used as an input to generate the frequent patterns and to compute the completeness. Experiments were run on a Dell XPS 27 with an Intel Core i7-4770S processor and 16GB of DDR3 RAM. The execution time of each experiment is insignificant (less than 5 seconds).

5.2 Impact of the number of instances

In this experiment we compare the completeness values, obtained for each category when varying the number of transactions (values range between 100 and 10000). Besides, we explain the completeness values that we obtained using a brute force approach (BFA), which measure the completeness according to the whole properties used in the descriptions of the instances. The results of DBpedia and Freebase categories are given in Figures 1 and 3.

We observe in these two figures that, concerning our metric, the obtained completeness values are relatively stable regarding the variation of the number of transactions (e.g. they range between 0.92 and 0.95 for the DBpedia category *Populated Place* with $\xi = 60\%$). For Freebase categories, the values are less regular than those of DBpedia. This means that the distribution of missing values in DBpedia is more regular than the one of Freebase. This observation is interesting as it expresses the fact that, for the used categories at least(e.g. *organization*), the instances have a widely divergent descriptions (despite the fact that they have the same type).

For the number of properties, represented in Figure 2 and 4, the conclusions are the same. The number of properties in $\hat{P}_w$ remains stable for the different itemsets sizes. This expresses a relative robustness of the metrics regarding the number of instances.

Concerning the completeness computed using the brute force approach, the values that we obtained are very low (less than 0.26 for both DBpedia and Freebase categories) and get lower (tend to zero) when the number of transactions grows (hence, we do not represent those values in the two figures). This is due to the fact that the instances from DBpedia or Freebase categories do not use all possible properties present in the selected set of instances. As we explained above in the motivating example, this approach could not provide us with a relevant value of completeness, as properties have not all the same importance.

5.3 Impact of the user-specified threshold ξ

To measure the impact of the minimum support threshold we rank correlation between ξ and the completeness, and between ξ and the number of properties in $\hat{P}_w$. We used for that the Spearman's rank correlation coefficient ρ.

[6]itemsets used in these experiments are available at: `http://cedric.cnam.fr/~hamdif/upload/cpmining2/`

ISSN 2429-4586

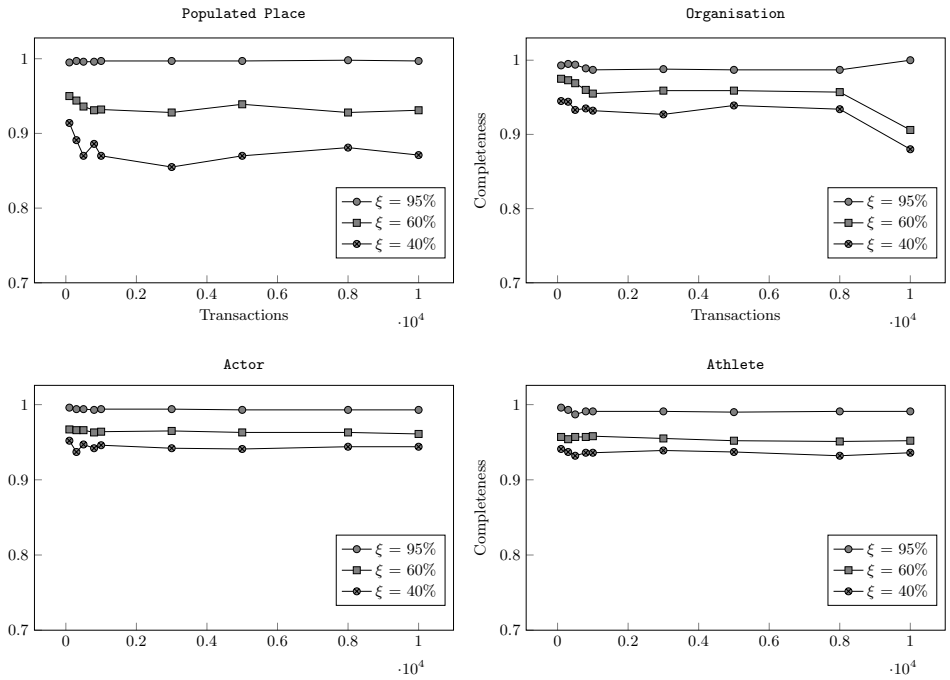

Figure 1: Completeness of DBpedia categories when varying the number of transactions and the minimum support ξ

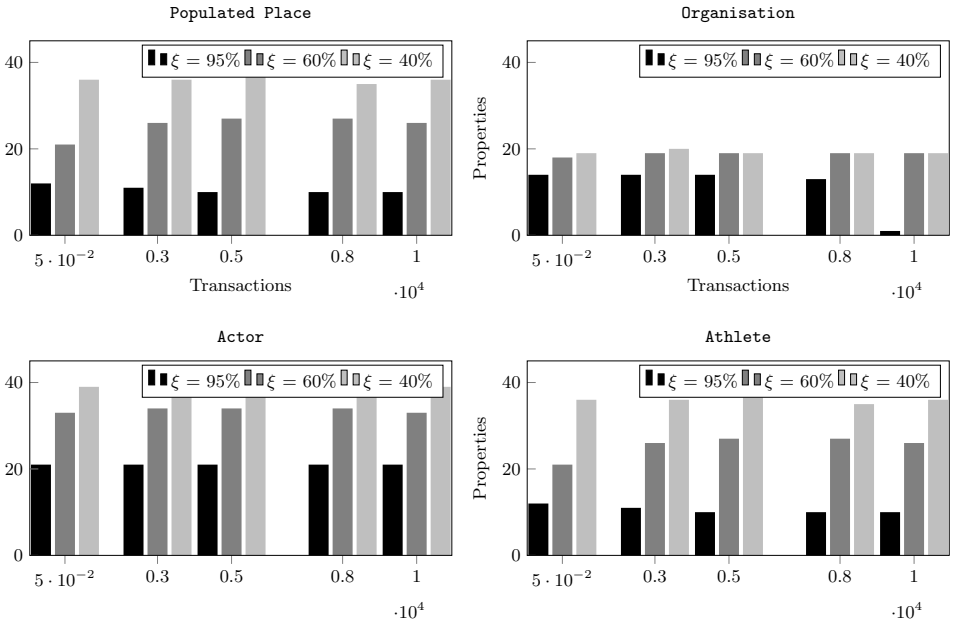

Figure 2: The number of properties in $\hat{P}_w$, for each DBpedia category, when varying the number of transactions and the minimum support ξ

ISSN 2429-4586

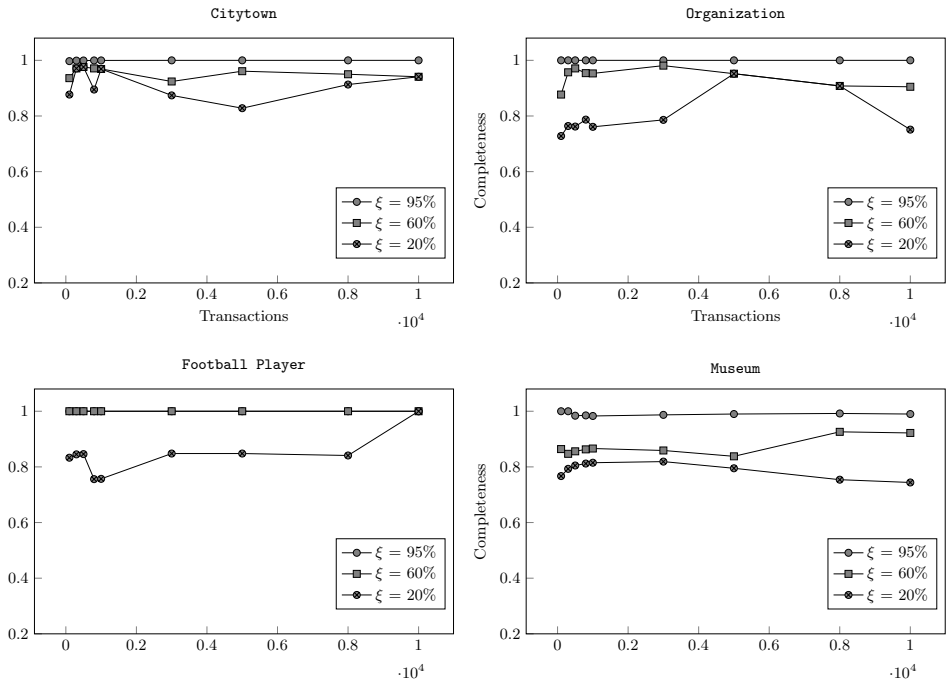

Figure 3: Completeness of Freebase categories when varying the number of transactions and the minimum support ξ

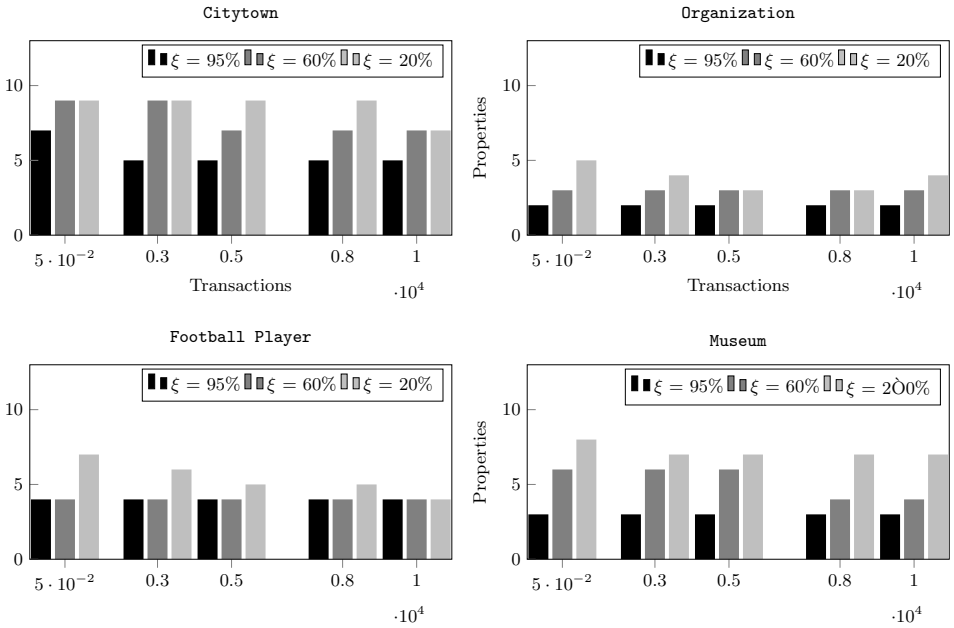

Figure 4: The number of properties in $\hat{P}_w$, for each Freebase category, when varying the number of transactions and the minimum support ξ

The values obtained for the sample of our experiments (by considering all DBpedia and Freebase categories, and all transactions sizes) are presented in the table 3.

To determine the significance of ρ, we used a t-distribution with $n-2$ degrees of freedom, and the standardized t statistic $t = \rho\sqrt{\frac{n-2}{1-\rho^2}}$ (where $n = 27$ is the size of our sample). For both DBpedia and Freebase categories , the positive correlation of ranks between ξ and the completeness is statistically significant at the 0.0005 level of significance (confidence level 99.9%), which is very well within the region of rejection of the null hypothesis. As a result, we have a large positive correlation between the user-specified threshold and the completeness. Concerning $|\hat{P}_w|$ a negative correlation with ξ seems to be obvious. Indeed, the negative correlation of the DBpedia categories is statically significant at the 0.005 level of significance. However, when we interpret ρ for the Freebase categories, we found that it is only significant at 0.05 (which still a good correlation). This is due to the fact that, as we can see in figure 4, there is no significant variation in the number of properties when ξ varies (especially for the *Football Player* category).

5.4 Discussion

Experiments that we carried out, show that the completeness metric provides meaningful values according to a pattern (or reference schema) mined from a dataset.

The first observation regarding the experiments concerns the behavior of the completeness metrics. The results obtained demonstrated the robustness of the proposed measure regarding the size of the dataset and even the nature of data. The higher stability in the values obtained for DBpedia compared to those of Freebase could be related to the fact that DBpedia has an underlying ontology that has been created based on the most commonly used infoboxes within Wikipedia. Freebase however, imports data from a wide variety of sources leading to less homogeneity in data descriptions. Moreover, we notice that values are better for categories such as *Actors* that are popular and probably attracts more publication effort than for *organization*.

The second observation is related to the user-defined threshold. This parameter refers to the desired expressiveness of the mined schema. The experiments showed a high correlation with the completeness of the datasets regarding this schema. This means that our approach is able to provide the user with the set of properties that ensures a degree of completeness meeting his/her requirements. Such a characteristics is very interesting and has several practical impacts. First, it provides a way to help expressing meaningful queries over RDF datasets when a schema describing the data source structure is missing. This schema, in addition to providing a query structure, ensures a degree of completeness that the user could control through the specification of the threshold. Note that the inferred pattern/schema, besides assuring good completeness values, has also a good expressiveness as it contains a significant number of properties (around 40 properties for some DBpedia categories). In the case of Freebase, the pattern is less expressive, as it is not greater than 12 (for considered categories in our experiments). So as a conclusion, we can say that, the more instances descriptions are homogeneous, the more completeness values are close (for any subset), and the more expressive are the inferred schemes.

Finally, once the completeness is measured regarding a reference schema for a given data source, the properties composing the schema could constitute good candidates for interlinking the dataset with external sources. Indeed, the completeness values obtained for the inferred schema will propagate through the links and assure a certain completeness for the external source.

6. RELATED WORKS

Information quality attracted many research works since two decades. It is an interesting theoretical as well as practical domain and several projects proposed methodologies to help dealing with quality assurance in traditional business information systems [21, 2, 4]. From the Web of Data perspective, the problem is a rather new issue. Researchers pointed out the fact that making data accessible is not sufficient especially when the users are companies and governmental agencies and especially when the target usage is business, research or countries' security. The credibility of data sources is thus bound to the quality of the content.

Several proposals could be classified using the 4 categories from the TDQM community [29] namely: *Intrinsic* (accuracy, reputation, believability and provenance), *Representational* (understandability, consistency and conciseness), *Accessibility* (accessibility and security), and *Contextual* (amount of data, relevance, completeness and timeliness). Intrinsic quality relies on internal characteristics of the data during evaluation. Most of the proposal concentrate on provenance data elicitation [25, 22, 15, 10]. From an external point of view, representational quality is concerned with factors influencing users interpretations and practices such as understandability [23] or data conciseness [30]. Concerning accessibility, a very important criteria in the context of web data, we could cite [18] where authors discussed common RDF publishers' errors that have a direct impact on data accessibility. Finally, contextual quality means that data could not be said "good" or "poor" without considering the context in which it is produced or used. To asses timeliness, the work presented in [16] relies on data provenance whereas authors in [5] propose metrics measuring the amount of data appropriate to meet associations mining requirements. The relevance dimension is considered from a ranking point of view in [8] and in the context of heterogeneous web data search in [17].

A quality criterion that remains critical and relies heavily on context is completeness. However, one can rarely find data having no missing entries. The incompleteness could have several reasons, ranging from human omission or misunderstanding during acquisition to data processing leading to lost data. The role of completeness evaluation is then to add information about data completeness to avoid biased analysis results and invalid conclusions. A first research direction focus on query answers completeness. We could cite [7] where authors introduced a framework to specify completeness statement on RDF data. To evaluate the completeness of data sources, Fürber at al. [9] distinguishes schema completeness from population or data completeness. The

Table 3: Spearman rank correlation ρ between ξ and $\mathcal{CP}$, and ξ and $|\hat{P}_w|$

$\mathcal{CP}$	$\lvert\hat{P}_w\rvert$	$\mathcal{CP}$	$\lvert\hat{P}_w\rvert$	$\mathcal{CP}$	$\lvert\hat{P}_w\rvert$	$\mathcal{CP}$	$\lvert\hat{P}_w\rvert$
Populated Place		Organisation		Actor		Athlete	
0.94	−0.82	0.9	−0.68	0.94	−0.81	0.94	−0.82
Citytown		Football Player		Museum		Organization	
0.81	−0.56	0.93	−0.73	0.82	−0.36	0.78	−0.81

former measures to which extent classes and properties are taken into account in data description. The second however, measures how complete is the population represented in the data compared to a complete population. A similar vision of completeness is adopted in [24] through extensional and intentional completeness. These definitions and the related metrics are very close to those used in the context of relational databases where the schema is the database schema and where the population refers to the database instances that is compared to an assumed "real world".

Note that the limitation of such a vision is that we should assume a closed-word assumption where a gold standard exists and could be used for both schema and population completeness assessment. However, our proposal, that addresses the schema completeness issue, do not rely on such an assumption, moreover difficult to meet. We propose to compute an intrinsic completeness value of the data source using both its extension (properties) and intention (instances). The underlying assumption is that, as data sources are populated by several persons and/or originate from several sources, the more frequent schema is likely to be a "consensus" schema.

7. CONCLUSION

We have presented in this article an approach for evaluating the completeness of RDF linked data sources. The approach is a two steps process that first computes a plausible schema as a set of properties and then computes the completeness of the related data source based on the discovered schema. Both schema and completeness computations use only the information from the data source without need of a gold standard.

The properties composing the schema are obtained by applying the well known FP-growth algorithm with an underlying assumption stating that a more frequent schema is likely to be more relevant. This solution is a compromise between the gold-standard approach difficult to apply in practice, and a brute force approach leading to irrelevant results.

The approach has been evaluated on two datasets that are DBpedia and Freebase. Several experiments have been conducted to evaluate the completeness measure robustness regarding the number of instances and the user-specified threshold. We have also analyzed how the approach behaves when varying several parameters factors. We have, for instance, studies the impact of the threshold and the number of transactions on the number of properties composing the mined schema.

Our analysis revealed some interesting characteristics allowing the characterization of the sources and the behavior of the community that maintains each of the data sources. The results show a low size of the computed schema for some categories such as *Organization* indicating the high heterogeneity of properties in the schema of data sources.

In the future, we will study how to improve the completeness of data sources by concentrating the effort on the more relevant properties. We will also work on how to use the completeness values to improve query expressions to increase the completeness of returned results.

References

[1] D. P. Ballou and H. L. Pazer. Modeling completeness versus consistency tradeoffs in information decision contexts. *Knowledge and Data Engineering, IEEE Transactions on*, 15(1):240–243, 2003.

[2] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino. Methodologies for data quality assessment and improvement. *ACM Computing Surveys (CSUR)*, 41(3):16, 2009.

[3] S. Bechhofer, I. Buchan, D. De Roure, P. Missier, J. Ainsworth, J. Bhagat, P. Couch, D. Cruickshank, M. Delderfield, I. Dunlop, et al. Why linked data is not enough for scientists. *Future Generation Computer Systems*, 29(2):599–611, 2013.

[4] L. Berti-Equille, I. Comyn-Wattiau, M. Cosquer, Z. Kedad, S. Nugier, V. Peralta, S. Si-said Cherfi, and V. Thion-Goasdoué. Assessment and analysis of information quality: a multidimensional model and case studies. *IJIQ*, 2(4):300–323, 2011.

[5] P. Chen and W. Garcia. Hypothesis generation and data quality assessment through association mining. In F. Sun, Y. Wang, J. Lu, B. Zhang, W. Kinsner, and L. A. Zadeh, editors, *Proceedings of the 9th IEEE International Conference on Cognitive Informatics, ICCI 2010, July 7-9, 2010, Beijing, China*, pages 659–666. IEEE, 2010.

[6] E. F. Codd. Missing information (applicable and inapplicable) in relational databases. *SIGMOD Record*, 15(4):53–78, 1986.

[7] F. Darari, W. Nutt, G. Pirrò, and S. Razniewski. Completeness statements about RDF data sources and their use for query answering. In H. Alani, L. Kagal, A. Fokoue, P. T. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. F. Noy, C. Welty, and K. Janowicz, editors, *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, volume 8218 of *Lecture Notes in Computer Science*, pages 66–83. Springer, 2013.

[8] C. M. Eastman and B. J. Jansen. Coverage, relevance, and ranking: The impact of query operators on web search engine results. *ACM Transactions on Information Systems (TOIS)*, 21(4):383–411, 2003.

[9] C. Fürber and M. Hepp. Swiqa-a semantic web information quality assessment framework. In *ECIS*, volume 15, page 19, 2011.

[10] J. Golbeck. Combining provenance with trust in social networks for semantic web content filtering. In L. Moreau and I. T. Foster, editors, *Provenance and Annotation of Data, International Provenance and Annotation Workshop, IPAW 2006, Chicago, IL, USA, May 3-5, 2006, Revised Selected Papers*, volume 4145 of *Lecture Notes in Computer Science*, pages 101–108. Springer, 2006.

[11] K. Gouda and M. J. Zaki. Efficiently mining maximal frequent itemsets. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 163–170, Washington, DC, USA, 2001. IEEE Computer Society.

[12] G. Grahne and J. Zhu. Efficiently using prefix-trees in mining frequent itemsets. In B. Goethals and M. J. Zaki, editors, *FIMI '03, Frequent Itemset Mining Implementations, Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, 19 December 2003, Melbourne, Florida, USA*, volume 90 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

[13] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In W. Chen, J. F. Naughton, and P. A. Bernstein, editors, *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA.*, pages 1–12. ACM, 2000.

[14] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Min. Knowl. Discov.*, 8(1):53–87, Jan. 2004.

[15] O. Hartig. Trustworthiness of data on the web. In *Proceedings of the STI Berlin & CSW PhD Workshop*. Citeseer, 2008.

[16] O. Hartig and J. Zhao. Using web data provenance for quality assessment. In J. Freire, P. Missier, and S. S. Sahoo, editors, *Proceedings of the First International Workshop on the role of Semantic Web in Provenance Management (SWPM 2009), collocated with the 8th International Semantic Web Conference (ISWC-2009), Washington DC, USA, October 25, 2009*, volume 526 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

[17] D. M. Herzig and T. Tran. Heterogeneous web data search using relevance-based on the fly data integration. In A. Mille, F. L. Gandon, J. Misselis, M. Rabinovich, and S. Staab, editors, *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, pages 141–150. ACM, 2012.

[18] A. Hogan, A. Harth, A. Passant, S. Decker, and A. Polleres. Weaving the pedantic web. In C. Bizer, T. Heath, T. Berners-Lee, and M. Hausenblas, editors, *Proceedings of the WWW2010 Workshop on Linked Data on the Web, LDOW 2010, Raleigh, USA, April 27, 2010*, volume 628 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010.

[19] M. G. Institute, M. Chui, J. Manyika, J. Bughin, R. Dobbs, C. Roxburgh, H. Sarrazin, G. Sands, and M. Westergren. *The social economy: Unlocking value and productivity through social technologies*. McKinsey Global Institute, July 2012.

[20] R. J. B. Jr. Efficiently mining long patterns from databases. In L. M. Haas and A. Tiwary, editors, *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA.*, pages 85–93. ACM Press, 1998.

[21] Y. W. Lee, D. M. Strong, B. K. Kahn, and R. Y. Wang. Aimq: a methodology for information quality assessment. *Information & management*, 40(2):133–146, 2002.

[22] M. Markovic, P. Edwards, D. Corsar, and J. Z. Pan. The crowd and the web of linked data: A provenance perspective. In *Wisdom of the Crowd, Papers from the 2012 AAAI Spring Symposium, Palo Alto, California, USA, March 26-28, 2012*, 2012.

[23] P. Mendes, C. Bizer, Y. Young, Z. Miklos, J. Calbimonte, and A. Moraru. Conceptual model and best practices for high-quality metadata.

[24] P. N. Mendes, H. Mühleisen, and C. Bizer. Sieve: linked data quality assessment and fusion. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, pages 116–123. ACM, 2012.

[25] T. Omitola, N. Gibbins, and N. Shadbolt. Provenance in Linked Data Integration. In S. Auer, S. Decker, and M. Hauswirth, editors, *Proc. of Linked Data in the Future Internet at the Future Internet Assembly, Ghent 16/17 Dec 2010*, volume 700 of *CEUR Workshop Proceedings ISSN 1613-0073*, February 2010.

[26] L. L. Pipino, Y. W. Lee, and R. Y. Wang. Data quality assessment. *Communications of the ACM*, 45(4):211–218, 2002.

[27] M. Samwald, A. Jentzsch, C. Bouton, C. S. Kallesøe, E. Willighagen, J. Hajagos, M. S. Marshall, E. Prud'hommeaux, O. Hassanzadeh, E. Pichler, et al. Linked open drug data for pharmaceutical research and development. *Journal of cheminformatics*, 3(1):19, 2011.

[28] Y. Wand and R. Y. Wang. Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, 39(11):86–95, 1996.

[29] R. Y. Wang and D. M. Strong. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, pages 5–33, 1996.

[30] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, S. Auer, and P. Hitzler. Quality assessment methodologies for linked open data. *Submitted to Semantic Web Journal*, 2013.

Interrogation par analogie dans les bases de données

Sara El Hassad
Université de Rennes 1 – IRISA/SHAMAN
sara.el-hassad@irisa.fr
Ce travail s'inscrit dans le projet PAWS partiellement financé par
Lannion-Trégor Communauté et la Région Bretagne

1. CONTEXTE ET MOTIVATIONS

L'*analogie*, notion bien connue en Sciences Cognitives et en Intelligence Artificielle, met en relation deux objets A et B sous la forme d'une assertion : « *A est comme B* » [7]. Elle se distingue de la similarité en mettant en avant les caractéristiques communes de deux objets, a priori différents, sur la base de leurs définitions. Ainsi, selon [7], un soleil est comme un noyau d'atomes car des objets gravitent autour d'eux (resp., des planètes et des électrons) : ce sont des centres de systèmes gravitationnels. La *proportion analogique* caractérise une analogie des relations entre une paire d'objets (A, B) d'une part, et entre une paire d'objets (C, D) d'autre part, sous la forme d'une assertion : « *A est à B ce que C est à D* » [10]. De l'analogie précédente, on tire la proportion analogique suivante : une planète est à un soleil ce qu'un électron est à un noyau d'atome.

L'objectif général de la thèse est d'étudier l'interrogation par analogie des bases de données. Nos contributions en cours et à venir sont la formalisation de ce nouveau type d'interrogation, l'énumération des problèmes de décision et de calcul associés et la résolution des challenges algorithmiques qui en découlent. Ces travaux doivent par exemple permetre à un journaliste de données, face à des données d'observations des écosystèmes naturels nationaux, de rechercher des analogies entre des espèces différentes. Celui-ci pourrait être amené à formuler des interrogations fondées sur l'analogie telles que : « Quelles espèces sont comme les algues vertes observées sur les côtes bretonnes ? » ; ou basées sur la proportion analogique liant les espèces à leur lieu de vie : « Quelles espèces sont aux rivières d'Aquitaine ce que les algues vertes sont aux côtes bretonnes ? »

La suite de cet article est organisée comme suit. Dans la section 2, nous formalisons ces relations en termes de notions classiques de Bases de Données et nous les illustrons. Dans la section 3, nous exposons les différents problèmes de décision et de calcul que nous allons étudier. Enfin, dans la section 4, nous pointons des travaux pouvant servir de bases à l'élaboration d'algorithmes pour résoudre ces problèmes.

(c) 2015, Copyright is with the authors. Published in the Proceedings of the BDA 2015 Conference (September 29-October 2, 2015, Ile de Porquerolles, France). Distribution of this paper is permitted under the terms of the Creative Commons license CC-by-nc-nd 4.0.
(c) 2015, Droits restant aux auteurs. Publié dans les actes de la conférence BDA 2015 (29 Septembre-02 Octobre 2015, Ile de Porquerolles, France). Redistribution de cet article autorisée selon les termes de la licence Creative Commons CC-by-nc-nd 4.0.
BDA 29 Septembre-02 Octobre 2015, Ile de Porquerolles, France.

2. ANALOGIE ET BASES DE DONNÉES

Nous formalisons ici les relations d'analogie et de proportion analogique dans un cadre général de bases de données.

Étant donné un alphabet de relations $\mathcal{R}$, une base de données est l'union d'un ensemble de contraintes $\mathcal{C}$ sur $\mathcal{R}$ et d'un ensemble de faits $\mathcal{F}$ sur $\mathcal{R}$. Ainsi, nous capturons le cadre des bases de données relationnelles lorsque $\mathcal{C}$ modélise des contraintes de dépendance fonctionnelle, d'inclusion, etc, ainsi que celui des bases de données ontologiques (ou bases de connaissances) quand $\mathcal{C}$ modélise des contraintes déductives à l'aide d'un dialect de logiques de description [2], de Datalog± [4] ou encore de Règles Existentielles [3]. Les objets impliqués dans les relations d'analogie et de proportion analogique sont des requêtes conjonctives sur $\mathcal{R}$.

Nous considérons deux objets comme étant en analogie, s'ils appartiennent à un même type plus général que le leur. Cette relation s'exprime naturellement dans notre cadre avec la notion d'*inclusion de requêtes* [1] : étant donné $\mathcal{C}$ sur $\mathcal{R}$, une requête Q' est incluse dans une requête Q, noté $Q' \subseteq_{\mathcal{C}} Q$, si et seulement si toutes les réponses de Q' sont des réponses de Q, pour tout $\mathcal{F}$ sur $\mathcal{R}$.

Définition 1 (Analogie). *Soit Q_A et Q_B deux requêtes sur $\mathcal{R}$. Q_A* est comme *Q_B ssi il existe une requête Q sur $\mathcal{R}$ telle que $Q_A \subseteq_{\mathcal{C}} Q$ et $Q_B \subseteq_{\mathcal{C}} Q$, où $\subseteq_{\mathcal{C}}$ est l'inclusion de requêtes modulo $\mathcal{C}$.*

Nous considérons que deux paires d'objets, (A, B) et (C, D), sont en proportion analogique si une relation entre A et B *est comme* une relation entre C et D. Ces relations entre objets s'expriment naturellement par des requêtes et leur analogie se traduit encore à l'aide de l'inclusion de requêtes.

Définition 2 (Proportion Analogique). *Soit Q_A, Q_B, Q_C et Q_D des requêtes sur $\mathcal{R}$. Q_A* est à *Q_B* ce que *Q_C* est à *Q_D ssi :*

- *il existe une requête Q_{AB} sur $\mathcal{R}$ telle que $Q_{AB} \not\subseteq_{\mathcal{C}} \bot$ et $Q_{AB} \subset_{\mathcal{C}} Q_A \times Q_B$,*
- *il existe une requête Q_{CD} sur $\mathcal{R}$ telle que $Q_{CD} \not\subseteq_{\mathcal{C}} \bot$ et $Q_{CD} \subset_{\mathcal{C}} Q_C \times Q_D$,*
- *il existe une requête Q sur $\mathcal{R}$ telle que $Q_{AB} \subseteq_{\mathcal{C}} Q$ et $Q_{CD} \subseteq_{\mathcal{C}} Q$.*

On remarquera que la définition ci-dessus impose aux requêtes exprimant des relations entre objets d'être *non-vides* et *sans* produit cartésien, afin de modéliser un lien sémantique réel entre les objets liés : sans cette contrainte, tout quadruplet d'objets formerait une proportion analogique.

Exemple (**Biodiversité**) Soit l'ensemble de contraintes $\mathcal{C} = \{\forall x(\texttt{AlgueVerte}(x) \Rightarrow \texttt{Espece}(x)),$

$\forall x(\texttt{Ecrevisse}(x) \Rightarrow \texttt{Espece}(x))$,
$\forall x(\texttt{AlgueVerte}(x) \wedge \texttt{Ecrevisse}(x) \Rightarrow \bot)$,
$\forall x(\texttt{BaieBretonne}(x) \Rightarrow \texttt{ZoneObservation}(x))$,
$\forall x(\texttt{RiviereAquitaine}(x) \Rightarrow \texttt{ZoneObservation}(x))$,
$\forall x \forall y(\texttt{prolifere}(x,y) \Rightarrow \texttt{envahit}(x,y))$,
$\forall x \forall y(\texttt{contamine}(x,y) \Rightarrow \texttt{envahit}(x,y))$,
$\forall x \forall y(\texttt{prolifere}(x,y) \Rightarrow \texttt{AlgueVerte}(x))$,
$\forall x \forall y(\texttt{prolifere}(x,y) \Rightarrow \texttt{BaieBretonne}(y))$,
$\forall x \forall y(\texttt{contamine}(x,y) \Rightarrow \texttt{Ecrevisse}(x))$,
$\forall x \forall y(\texttt{contamine}(x,y) \Rightarrow \texttt{RiviereAquitaine}(y))\}$.

Les objets $Q_A(x){:-}\texttt{AlgueVerte}(x) \wedge \texttt{prolifere}(x,y) \wedge \texttt{BaieBretonne}(y)$ et $Q_B(x){:-}\texttt{Ecrevisse}(x) \wedge \texttt{contamine}(x,y) \wedge \texttt{RiviereAquitaine}(y)$ sont en analogie car la requête $Q(x)$: $-$ $\texttt{Espece}(x) \wedge \texttt{envahit}(x,y) \wedge \texttt{ZoneObservation}(y)$ les inclut.

Les objets $Q_A(x){:-}\texttt{AlgueVerte}(x)$, $Q_B(x){:-}\texttt{BaieBretonne}(x)$, $Q_C(x){:-}\texttt{Ecrevisse}(x)$ et $Q_D(x){:-}\texttt{RiviereAquitaine}(x)$ sont en proportion analogique car il existe $Q_{AB}(x,y){:-}\texttt{AlgueVerte}(x) \wedge \texttt{prolifere}(x,y) \wedge \texttt{BaieBretonne}(y)$ qui lie les algues vertes aux baies Bretonnes, $Q_{CD}(x,y){:-}\texttt{Ecrevisse}(x) \wedge \texttt{contamine}(x,y) \wedge \texttt{RiviereAquitaine}(y)$ qui lie les écrevisses américaines aux rivières d'Aquitaine et $Q(x,y){:-}\texttt{Espèce}(x) \wedge \texttt{envahit}(x,y) \wedge \texttt{ZoneObservation}(y)$ qui les inclut. $\diamond$

3. PROBLÈMES À ÉTUDIER

Sur la base du cadre défini ci-dessus, nous nous intéressons à l'algorithmique de problèmes de décision et de calcul.

Les problèmes de décision consistent, étant donné un ensemble de contraintes $\mathcal{C}$, à dire – Oui ou Non – si deux objets sont en analogie ou si deux paires d'objets sont en proportion analogique. Ceci revient à *décider* s'il existe une requête Q au sens de la Définition 1 pour l'analogie ou de la Définition 2 pour la proportion analogique. Les problèmes de calcul consistent (i) à expliquer et (ii) à découvrir des analogies ou des proportion analogiques. Expliquer une analogie revient, pour une instance positive d'un problème de décision décrit ci-dessus, à *exhiber* une requête Q susmentionnée. Découvrir une analogie ou proportion analogique consiste à trouver les *instances positives* de problèmes de décision décrits ci-dessus, dont les objets sont partiellement connus. Ceci permet de poser des questions telles que : Quel x est comme Q_B ? Quel x est à Q_B ce que Q_C est à Q_D ? Etc.

Les challenges algorithmiques sous-jacents à la résolution de ces problèmes sont doubles. Tout d'abord, il s'agit de *calculer* des généralisants communs de requêtes modulo un ensemble de contraintes, en particulier des plus petits au regard de la relation d'inclusion de requêtes. Les généralisants communs permettent de caractériser l'existence de requêtes Q au sens de la Définition 1 et de la Définition 2 ; les plus petits généralisants communs permettent de les expliquer de façon la plus intelligible possible à un utilisateur, car ils sont plus proches sémantiquement des l'énoncés des problèmes. Pour calculer les requêtes Q_{AB} et Q_{CD} au sens de la Définition 2, nous faisons appel à la spécialisation de requêtes modulo un ensemble de contraintes. Ensuite, pour les problèmes autres que la décision et l'explication d'analogie, il s'agit de chercher des requêtes (objets ou relations entre eux) caractérisant une analogie ou proportion analogique. On notera que l'espace de recherche est particulièrement grand. Par exemple, dans le cas le plus simple, chercher des x comme un Q_A donné, consiste à trouver toutes les requêtes incluses dans toutes celles généralisant Q_A ! Il faudra donc restreindre l'espace effectif de recherche avec des métriques sur la relation d'inclusion éventuellement complétée par des statistiques sur les données interrogées, afin d'éviter les généralisations et spécialisations excessives.

4. TRAVAUX EN COURS

Notre cadre d'interrogation par analogie pour les bases de données ontologiques utilise la logique de description « légère » DL-lite$_\mathcal{R}$ ([5]) comme langage de contraintes. Ce fragment pragmatique de Datalog± et des Règles Existentielles fournit les fondements d'OWL2 QL, un modèle de données sémantiquement riche du W3C. Actuellement, nous nous focalisons sur le calcul, fondamental à la mise en place du cadre d'interrogation proposé, de généralisants et en particulier de plus petits généralisants communs. À notre connaissance, ce problème n'a pas été étudié pour DL-lite$_\mathcal{R}$, contrairement au problème dual de calcul de requêtes incluses dans une requête donnée qui est une composante nécessaire pour répondre à une requête par reformulation (par ex. [5, 8]). Nous nous inspirons donc de résultats algorithmiques issus de différents domaines de l'informatique dans lesquels le problème du calcul d'un plus petit généralisant commun a été traité pour des formalismes logiques proches. En particulier, ce problème admet une solution (trop) générale pour les *clauses de Horn sans symbole de fonction* en Programmation Logique Inductive [9], ainsi qu'une solution (trop) spécifique pour les *graphes conceptuels simples* en Représentation des Connaissances [6]. Nos algorithmes se fonderont sur ces travaux afin de proposer une solution adaptée et optimisée pour DL-lite$_\mathcal{R}$.

5. REFERENCES

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge Univ. Press, 2003.

[3] J. Baget, M. Leclère, M. Mugnier, and E. Salvat. On rules with existential variables : Walking the decidability line. *Artificial Intelligence*, 175, 2011.

[4] A. Calì, G. Gottlob, and T. Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. In *PODS*, 2009.

[5] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics : The *DL-Lite* family. *JAR*, 39(3) :385–429, 2007.

[6] M. Chein and M.-L. Mugnier. *Graph-based Knowledge Representation : Computational Foundations of Conceptual Graphs*. Springer, 2008.

[7] D. Gentner. Structure-mapping : A theoretical framework for analogy. *Cognitive Science*, 7(2), 1983.

[8] G. Gottlob, G. Orsi, and A. Pieris. Query rewriting and optimization for ontological databases. *ACM TODS*, 39(3), 2014.

[9] S.-H. Nienhuys-Cheng and R. de Wolf. Least generalizations and greatest specializations of sets of clauses. *J. of Artificial Intelligence Research*, 1996.

[10] H. Prade and G. Richard, editors. *Computational Approaches to Analogical Reasoning : Current Trends*, volume 548 of *Studies in Computational Intelligence*. Springer, 2014.

Scalable Approaches for Recommendation in Social Networks

Yifan LI*
LIP6, Université Paris 6
Paris, France
yifan.li@lip6.fr

ABSTRACT

Nowadays, the online social network has become a significant part of our life, and deeply influenced our activities in many aspects than one might imagine before. Hereby, with the benefit from its increasing growing, there are more prediction opportunities emerging for recommendation system deployment. Rather than as only a supplementary of the traditional use of *collaborative filtering* (CF) method, in some cases[7], the social affinity among users can provide more precision in recommendation result(scores) calculation.

In this doctoral thesis, we propose to design a scalable recommendation approach over large social graph, taking into consideration not only graph topological properties but also those user semantic content, e.g. user interest, (hash-)tags, item ratings, to gain a good and fast recommendation evaluation.

Keywords

Social Network, Recommendation System, Graph, Pregel, Personalized PageRank

1. INTRODUCTION

Social network, e.g. Facebook, Linkedin and Weibo, as a new but sharply growing media, has permeated our lives. For instance, by April 2014, the main micro-blogging service in the world, Twitter, has build a social graph consisting of more than 570 million users. Moreover, around one million new accounts are registered in Twitter every week, and 500 million fresh tweets are posted every day. Hence, this rapidly increasing data volume and natural structure dynamism present twofold serious challenges to our existing data management system, 1) *distributed* graph storage and computation and 2) *scalable* algorithm performing.

To deal with the first problem: most of social graphs are too large to fit in memory and disk on a generic machine, several distributed graph computation models, such as Pregel [6], PowerGraph[4], GraphX [8], have been proposed to meet this requirement. The graph partitioning strategy study, however, has *not* been fully conducted in these systems in which it is indeed the basis for distribution. For this reason, our first task is to propose an efficient flexible partitioning approach for processing large social graphs in a distributed way, in stead of a simple *hash function* method with *high communication cost*.

For answering the query for a given node in social graph, we need to evaluate the calculation of *topological similarity* between two nodes in graph. Whereas the volume of the social network and its dramatical growth raise scalability issue. However these existing graph algorithms don't give sufficient weight on either increasing *computation parallelism* or exploring *topological feature* inside graph partition. Thus, in my thesis we plan to propose an segments-based similarity computation algorithms that can be implemented on large scale graph processing systems that offer transparent scalability, high computation parallelism and fully using of graph partitioning feature.

2. SCALABLE GRAPH PROCESSING

2.1 Graph Partitioning

Naturally, how to partition a social network is the foundation of a distributed graph processing system. Given a graph application, the partitioning strategy has a huge impact on its performance. From perspective of querying, a good graph partitioning also can advance the query evaluation in a data/computation-distributed context.

Hence, here we proposed a novel workload-aware *Block-based Partitioning Strategy* to acquire targets: 1) Minimised Communication, 2) Balanced Workload

Block Construction using Seeds. In this thesis, a block can be seen as one, or part of tightly knit cluster in graph, e.g. a community in social network, which is formed by several edges, rather than vertices. We propose to find those K inherent blocks B_1, B_2, ..., B_K from graph G by exploring its full potential topology. Especially, there are K pre-selected seed vertices, of which one is per block as its centre. Then we compute the closenesses, from seeds, to the two nodes of every edge, which will be used to determine the ownership of edge finally.

Closeness between Nodes. Since the block an edge belongs is determined by the closeness(namely reachability) from its seed(center) to that edge's two end-nodes. Firstly we propose to offer a method to measure that closeness, by

*supervised by Prof. Cedric du Mouza(CNAM, Paris) and Prof. Camelia Constantin(LIP6-UPMC, Paris)

(c) 2015, Copyright is with the authors. Published in the Proceedings of the BDA 2015 Conference (September 29-October 2, 2015, Ile de Porquerolles, France). Distribution of this paper is permitted under the terms of the Creative Commons license CC-by-nc-nd 4.0.
(c) 2015, Droits restant aux auteurs. Publié dans les actes de la conférence BDA 2015 (29 Septembre-02 Octobre 2015, Ile de Porquerolles, France). Redistribution de cet article autorisée selon les termes de la licence Creative Commons CC-by-nc-nd 4.0.
BDA 29 September 2015, Porquerolles, France.

using a simpler *inverse P-distance*:

$$D_i(j) \equiv \sum_{p \in \{i \sim j\}} S(p)$$

where the p is a path from node i to node j or vice versa, which is formed by a sequence of edges. *S(p)* is the inverse distance value of path p: v_0, v_1, ..., $v_{(k-1)}$ with length k, and is defined as

$$S(p) = (1-\alpha)^k \cdot \prod_{i=0}^{k-1} \frac{1}{outDeg(v_i)}$$

where $\alpha \in (0,1)$is the teleporting probability in the random walk, which means the probability to return origin. $outDeg(v_i)$ is the out-degree of vertex v_i.

Edge Allocation. For each edge triplet, we will calculate the two closeness-score vectors together to obtain the final result: the block(seed) id it belongs.

Block Refinement. Due to the limits for workload balancing, or computation balancing further, we need to conduct some kind of necessary refinement operations on blocks, e.g. to combine several small ones or split a large one.

Demostration. Up to now, from our existing experiments, the communication measurement[4], *vertex replication factor*, has been reduced to around *40 percent*, and it has also cut the graph processing time in half, compared with the EdgePartition2D in [8] which has best performance generally. Then, we tend to perform more experiments to observe boundary-crossed communication between partitions, esp. for our segments-based algorithm.

In addition, we will further give a fresh analysis on *computation balancing*, rather than storage balancing, for the first time.

2.2 Scalable Similarity Algorithm

The main target for recommendation on a social network is to find the most *similar* node(s) for a given node. This task can be considered from two aspects: 1) searching the topologically close node(s) and 2) searching the semantic-close node(s).

Among those existing graph-based similarity computation algorithms, e.g. Hitting Time, PageRank, Graph Distance, we made an adapted segments-based Monte Carlo method to approximate Fully Personalized PageRank(PPR) to measure the topological *similarity* between nodes [1, 5].

Here, as the ideas in [3, 2], instead of simulating a complete long random walk over the whole graph, several short walks(of lengths geometrically distributed on teleporting factor) are conducted starting from the common node. Finally the approximate PPR value can be obtained by counting the frequencies of visits to each node in short walks. Especially, still, each short random walk can be divided into shorter segments, thus in this way we can increase the parallelism of graph computation which is implemented in Pregel model. Moreover, the considerable short segments(random walk) starting from the same initial node are expected to stay in a common graph partition.

3. FUTURE WORK

3.1 Seeds Selection Strategy

The quality of seeds has a big effect on graph partitioning result, for this reason, we need to make more effort to seek out a way for adaptive seed selection with various graph topologies.

3.2 Personalized PageRank

We have had a straight-forward segments-based random walk simulation algorithm that can be applied to PPR easily, but nonetheless, more optimisations, w.r.t Pregel computation model, are still worthwhile.

3.3 Graph Evolving

The dynamism is another big challenge in computing large graphs of real world, social network etc. We plan to extend the recommendation model to deal with the change of social network structure and user content, with necessary but acceptable additional cost.

4. REFERENCES

[1] K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova. Monte carlo methods in pagerank computation: When one iteration is sufficient. *SIAM J. Numer. Anal.*, 45(2):890–904, Feb. 2007.

[2] A. Das Sarma, S. Gollapudi, and R. Panigrahy. Estimating pagerank on graph streams. In *Proceedings of the Twenty-seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '08, pages 69–78, New York, NY, USA, 2008. ACM.

[3] D. Fogaras and B. RÃącz. Towards scaling fully personalized pagerank. In *In Proceedings of the 3rd Workshop on Algorithms and Models for the Web-Graph (WAW*, pages 105–117, 2004.

[4] J. E. Gonzalez, Y. Low, H. Gu, D. Bickson, and C. Guestrin. Powergraph: Distributed graph-parallel computation on natural graphs. In *Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation*, OSDI'12, pages 17–30, Berkeley, CA, USA, 2012. USENIX Association.

[5] G. Jeh and J. Widom. Scaling personalized web search. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 271–279, New York, NY, USA, 2003. ACM.

[6] G. Malewicz, M. H. Austern, A. J. C. Bik, J. C. Dehnert, I. Horn, N. Leiser, G. Czajkowski, and G. Inc. Pregel: A system for large-scale graph processing. In *In SIGMOD*, pages 135–146, 2010.

[7] A. Olteanu, A.-M. Kermarrec, and K. Aberer. Comparing the predictive capability of social and interest affinity for recommendations. In B. Benatallah, A. Bestavros, Y. Manolopoulos, A. Vakali, and Y. Zhang, editors, *Web Information Systems Engineering âĂŞ WISE 2014*, volume 8786 of *Lecture Notes in Computer Science*, pages 276–292. Springer International Publishing, 2014.

[8] R. S. Xin, J. E. Gonzalez, M. J. Franklin, and I. Stoica. Graphx: A resilient distributed graph system on spark. In *First International Workshop on Graph Data Management Experiences and Systems*, GRADES '13, pages 2:1–2:6, New York, NY, USA, 2013. ACM.

Interactive Rule Mining in Knowledge Bases

Luis Galárraga
Télécom ParisTech
Paris, France
galarrag@enst.fr

ABSTRACT

L'extraction de règles d'association dans une base de connaissance consiste en la découverte de façon automatique de règles logiques à partir de faits connus. Cette tâche reste cependant un défi pour plusieurs raisons. D'abord, les bases de connaissances ne contiennent pas d'informations négatives, c'est-à-dire, les méthodes d'extraction de règles ne disposent pas de contre-exemples. En outre, les bases de connaissance existantes sont volumineuses avec des millions de faits ; cela accroit l'espace de recherche de manière considérable. Les systèmes d'extraction de règles actuels réduisent l'espace de recherche par une exclusion systématique des candidats peu prometteurs pendant le processus de construction des règles.

Dans cet article, nous proposons un processus d'extraction de règles transparent à l'utilisateur au travers un modèle de démonstration interactif.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Rule Mining, Knowledge Bases

1. INTRODUCTION

A knowledge base (KB) is a collection of machine-readable information designed to model a certain domain of knowledge. Initiatives such as YAGO [19], DBpedia [2], Wikidata[1], and the Knowledge Vault [7], among others, have constructed large-scale KBs about people, places, organizations, etc., by applying Information Extraction (IE) techniques on the Web [8, 19, 6, 3]. KBs are also prominent in the Life Sciences, with projects such as BioRDF[2] or UniProt[3]. KBs are crucial when computers have to process natural language text in a semantic fashion. The Knowledge Vault, for instance, allows the Google search engine to identify concepts and proper names in the words of queries. This facilitates the retrieval of answers about real-world entities instead of pages matching the words in the query.

KBs can be mined for patterns and rules. For instance, we could learn that Grammy awardees frequently play guitar. We could also find out that usually, people live in the same place as their spouses. If we resort to a datalog-like notation of facts, these examples can be formulated as logical rules:

$$musicalRole(x, Guitar) \Leftarrow wonPrize(x, Grammy) \quad (1)$$

$$livesIn(x, y) \Leftarrow isMarriedTo(z, x) \wedge livesIn(z, y) \quad (2)$$

These rules provide not just insight by themselves, they can also be used to predict information or propose corrections to the data. For example, assume that the KB knows *livesIn(Barack Obama, Washington)* and *marriedTo(Barack Obama, Michelle)* but does not know the place of residence of *Michelle*. Then we can use the second rule to deduce *livesIn(Michelle, Washington)*, that is, rules can be used to deduce new information [10]. We can also use the rules to correct erronous facts. If, say, the KB states that Michelle lives in a different city, then this statement can be flagged as suspicious. Other applications of rules include data maintenance tasks such as schema inference [20], KB alignment [11], or KB canonicalization [9].

Rule mining on KBs is a challenging endeavor for multiple reasons. First, today's KBs scale up to millions of facts and entities, and thus the search space for rule mining is huge. Second, Web-extracted KBs often contain gaps and noise that hinder the discovery of frequent and interesting patterns. Third, while mining logical rules on structured data has been widely studied by the Inductive Logic Programming (ILP) community, traditional ILP methods require explicit counter-examples. KBs, in contrast, operate under the *Open World Assumption* (OWA) and do not represent negative information.

AMIE [10] is a system that tackles the aforementioned challenges. The system is designed to mine rules on large KBs under the OWA. This is possible thanks to a novel confidence metric that gauges the quality of a rule even in the absence of counter-examples. Furthermore, AMIE is very efficient, i.e., it can mine rules on KBs with millions of facts in a matter of minutes. The key of AMIE's scalability is its tailored data-storage implementation and its wide range of pruning techniques, which cut out unpromising rules early on in the mining. These techniques include thresholding on

[1] `http://www.wikidata.org`
[2] `http://bio2rdf.org/`
[3] `http://www.uniprot.org/`

(c) 2015, Copyright is with the authors. Published in the Proceedings of the BDA 2015 Conference (September 29-October 2, 2015, Ile de Porquerolles, France). Distribution of this paper is permitted under the terms of the Creative Commons license CC-by-nc-nd 4.0.
(c) 2015, Droits restant aux auteurs. Publié dans les actes de la conférence BDA 2015 (29 Septembre-02 Octobre 2015, Ile de Porquerolles, France). Redistribution de cet article autorisée selon les termes de la licence Creative Commons CC-by-nc-nd 4.0.
BDA 29 septembre 2015, Ile de Porquerolles, France.

support and confidence values, exploiting monotonicity, and a lookahead on the next step.

To the user, however, the rule mining process appears as a black box: In its default setting, AMIE does not take any parameters or language bias as input, and so the user just points AMIE to the KB file and lets her run. AMIE takes some minutes to load the data and compute the rules, and then outputs the result. Thus, the user does not actually understand how these rules are mined or why certain rules were not mined.

With this demo, we aim to shed light on the inner workings of AMIE. We want to show the user why certain rules are mined, for what reason certain others are discarded, and how AMIE maneuvers in a search space that appears near infinite to the user. Our demo allows the user to "join" AMIE live in the mining process, and to see and influence the decisions that AMIE makes.

2. RELATED WORK

Several systems have been developed for rule mining [21, 17, 4, 12, 15, 5, 14, 16] on KBs. None of them, however, is explicitly designed to mine rules on large and potentially incomplete KBs under the OWA as AMIE. WARMR [12] is a system that reconciles ILP and association rule mining. In the spirit of traditional association rule mining, WARMR operates under the Closed World Assumption (CWA), that is, if a rule makes a prediction that is not present in the KB, the system counts it as a counter-example for the rule. ALEPH [14] is another ILP system, which offers a suite of different metrics to evaluate the quality of rules. One of them is the positives-only learning function, which does not require explicit counter-examples, but evaluates rules based on their length and the number of random facts deduced by the rule. In [10], it is shown that AMIE's confidence metric outperforms the positives-only learning function in terms of precision, when the rules are used for data inference. Quickfoil [21] is a traditional ILP system, which unlike WARMR or ALEPH, scales to millions of facts. Quickfoil, on the other hand, requires the user to provide explicit negative information for rule induction.

There have been also some work on unveiling the internals of rule mining to the users, and showing its applications. Crowdminer [1] is an interactive system that applies association rule mining techniques on data collected from the crowd. The system combines a proactive crowdsourcing approach to ask the users the most *benefitial* questions (those that maximize the knowledge gain) about their well-being practices such sports or dietary habits. The system allows the user to navigate through the most significant association rules and their metrics, e.g., for Crowdminer, a transaction is a bag of terms and rule looks like $\{coffee\} \Leftarrow \{headache\}$. SPIRE [13] is a tool targeted to data analysts that allows for intuitive visualization of the rule mining search space. One particularity of SPIRE is its support for negative conditions in rules, e.g., in our context rules such as $\neg deathPlace(x, z) \Leftarrow is(x, LivingPerson)$. The system can help data scientists identify the rules and correlations that are relevant to a particular application or scenario, in particular when rules are numerous.

3. PRELIMINARIES

Logical rules are built on *atoms*. An atom is of the form $r(x, y)$, where r is a relation x and y are either entities (such as *Barack Obama*) or variables (such as a). At least one of the arguments has to be a variable. A *Horn rule* is an expression of the form

$$r(x, y) \Leftarrow \boldsymbol{B}$$

where $r(x, y)$ is the *head* or *conclusion* of the rule, and $\boldsymbol{B}$ is a sequence of atoms $r_1(x_1, y_1), \ldots, r_m(x_m, y_m)$ called the *body*. Two atoms are connected if they share at least one variable. A rule is *connected* if every atom in the rule is transitively connected to every other atom. A rule is *closed* if every variable appears in at least two different atoms.

Rules encode patterns that occur frequently in the data. This notion of frequency is measured by the *support* of the rule. Support is calculated as

$$supp(r(x, y) \Leftarrow \boldsymbol{B}) := \#(x, y) : \exists z_1, ..., z_m : r(x, y) \wedge \boldsymbol{B}$$

Here, $\#(x, y) : \Phi$ are the number of pairs (x, y) that fulfill the condition Φ in the KB. A normalized version of the support is called the *head coverage* and is defined as:

$$hc(r(x, y) \Leftarrow \boldsymbol{B}) := \frac{supp(r(x, y) \Leftarrow \boldsymbol{B})}{\#(x, y) : r(x, y)}$$

The head coverage normalizes the support of the rule over the size of the head relation. This is convenient for KBs with small relations that would be ignored even for reasonable support thresholds. Support and head coverage are monotonic metrics, i.e., the addition of new atoms to a rule will never increase its support. This property is crucial for the AMIE algorithm. Support and head coverage are measures of statistical evidence. However, they do not gauge the *precision* of the rule. This dimension is taken into account by the standard confidence metric:

$$stdconf(r(x, y) \Leftarrow \boldsymbol{B}) := \frac{supp(r(x, y) \Leftarrow \boldsymbol{B})}{\#(x, y) : \exists z_1, ..., z_m : \boldsymbol{B}}$$

The standard confidence normalizes the support of the rule by the number of bindings of the head variables that match the antecedent of the rule. In other words, it measures the ratio of correct conclusions drawn from the rule. The standard confidence operates under the Closed World Assumption (CWA): It counts as counter-evidence all the conclusions of the rule that do not appear in the KB. However, under the OWA, conclusions may be true even if they are not in the KB. Hence, AMIE proposes an alternative confidence metric, as we shall see.

4. THE AMIE SYSTEM

4.1 Functions

A *function* is a relation that maps each subject to at most one object, such as *place of birth*. A relation is an *inverse function*, if its inverse is a function. Due to the noise in Web-extracted KBs, it is rare to find conceptually functional relations that comply strictly with these definitions. Therefore, [18] defines a functionality score for relations:

$$fun(r) := \frac{\#x : \exists y : r(x, y)}{\#(x, y) : r(x, y)}$$

The functionality is 1 for functional relations, and decreases towards 0 for less functional relations. The inverse functionality of a relation is defined as the functionality of its inverse

relation $ifun(r) := fun(r^{-1})$. AMIE assumes that relations are more functional than inverse functional ($fun(r) \geq ifun(r)$). If it is not the case, a relation r can always be replaced by its inverse relation r^{-1}. This is called the FUN-property of KBs in [10].

4.2 PCA Confidence

The Partial Completeness Assumption (PCA) [10] assumes that if a KB knows *some r-values* for an entity, then it knows *all its r values* for that entity.[4] For example, if the KB knows 4 official languages for Switzerland, then the PCA assumes than Switzerland has no further official languages. If a rule predicts a new language, the PCA labels the conclusion as false. If, in constrast no official language for Switzerland is known, then the PCA will not assume anything, i.e., it will label any conclusion for the language of Switzerland as unknown. The CWA, in contrast, would label both conclusions as false.

This means that the PCA admits that the KB can be incomplete. Therefore, it is more suitable for rule mining under the OWA. Under the PCA, we compute the confidence of a rule as follows:

$$pcaconf(r(x,y) \Leftarrow \boldsymbol{B}) := \frac{supp(r(x,y) \Leftarrow \boldsymbol{B})}{\#(x,y) : \exists z_1, \ldots, z_k, y' : r(x,y') \wedge \boldsymbol{B}}$$

The PCA confidence normalizes the support of the rule over the number of positive cases and the cases that the PCA assumes to be false. Experiments [10] suggest that the PCA confidence is more effective than the standard confidence at identifying productive rules, i.e., rules that produce many correct conclusions beyond the ones known to the KB.

4.3 Algorithm

The AMIE algorithm explores the search space of closed Horn rules by a breadth-first search. The algorithm starts with a queue that contains only the empty rule (with no head and no body). In each iteration, the algorithm dequeues a rule and applies one of the following mining operators:

- **Add Dangling Atom** ($\mathcal{O}_\mathcal{D}$)
  This operator adds a new atom to a rule. The new atom uses a fresh variable for one of its two arguments. The other argument (variable or entity) is shared with the rule, i.e., it occurs in some other atom of the rule.
- **Add Instantiated Atom** ($\mathcal{O}_\mathcal{I}$)
  This operator adds a new atom that uses an entity for one of its arguments, and shares the other argument (variable or entity) with the rule.
- **Add Closing Atom** ($\mathcal{O}_\mathcal{C}$)
  This operator adds a new atom that shares both of its arguments with the rule.

The new, refined, rules are then added to the queue. This process is run in multiple threads until the queue is empty. AMIE relies on a tailored in-memory triple store, which is optimized for the kind of queries that the operators generate. This component in combination with the pruning (s.b.) allows AMIE to mine closed Horn rules on KBs three orders of magnitude faster than other state of the art systems such as WARMR [12] or ALEPH [15].

[4]The PCA was later used as the *local closed world assumption* in [7].

4.4 Pruning

If run naively, the AMIE algorithm would enumerate the entire search space. Thus, it would not finish in reasonable time. Therefore, AMIE limits the search as follows:

1. By default, AMIE mines rules only up to 3 atoms. The user can choose to increase this threshold, but experiments [10] show that longer rules capture only very obscure correlations in practice.
2. If a rule has head coverage below 1%, AMIE considers it too insignificant, and filters it out. However, the user can also provide a different threshold value or prune on absolute support instead.

Since support, head coverage, and the length of the rule are monotonic metrics, AMIE can safely abandon any further refinement of such rules as well.

In addition, AMIE uses the following pruning techniques:

1. If the rule is closed and has a PCA confidence of 1.0, then no refinement can make the rule better, because confidence cannot increase and support can only decrease. Hence, the rule is output and removed from the queue. We call these rules *perfect rules*.
2. If the new rule has a lower confidence than the rule that it is derived from, then the rule is just silently enqueued for further refinement, but not output. This is because it will have lower confidence *and* lower support than the previous rule. This technique is called the *skyline technique* and was designed to avoid over-specifications of the same logical rule.
3. If the maximum permitted length of a rule is n, then AMIE will not apply the operator $\mathcal{O}_\mathcal{D}$ at length $n-1$. This is because the operator adds a new variable, and thus the rule cannot be closed within the limit of n atoms. This technique is called the *lookahead*.
4. When AMIE adds a new instantiated atom, it will exclude atoms where the variable of this atom can bind to only one entity in the KB. This is because these cases induce a *quasi-binding* of the free variable, meaning that there is a shorter equivalent rule. For example, it does not make sense to add the atom *hasOfficialLanguage(x,Romansh)* to a rule, because x can bind only to *Switzerland*. Thus, there is a shorter equivalent rule, namely the one where the atom is not added and x is replaced by *Switzerland*.

5. DEMO

5.1 Overview

Our demo lets the user understand the heuristics that govern AMIE's search strategy by allowing the user to navigate through the search space together with AMIE. For this purpose, our demo shows the rule construction process, starting from the empty rule. At each step, the user can add a new atom $r(x,y)$. This involves two steps: (1) choosing the relation r and (2) choosing the arguments x, y. The second step offers various permutations of arguments: Each argument can be a variable (either known or fresh) or an entity (chosen from the set of entities that lead to the highest support).

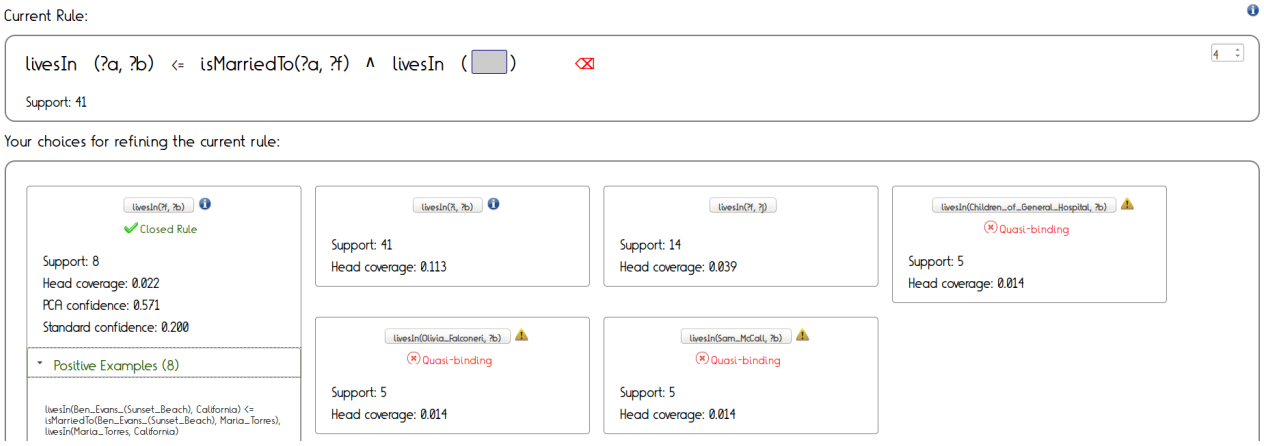

Figure 1: Mining a closed Horn rule

For instance, imagine the user wants to follow the path to mine the rule

$$livesIn(a, b) \Leftarrow isMarriedTo(a, f) \wedge livesIn(f, b)$$

The demo starts with the empty rule "... $\Leftarrow$ ..." and asks the user to select a relation for the head atom. The system shows as choices a list of relations ranked by size. Once the user picks a relation (i.e., *livesIn* in our example), the system applies the mining operators on the empty rule and reports all the refinements that have *livesIn* as relation, ranked by support. This includes, for example, the atom $lives(a, b)$ from the operator $\mathcal{O}_{\mathcal{D}}$, or the atom $livesIn(a, California)$ from the operator $\mathcal{O}_{\mathcal{I}}$. When applied the first time, the operator $\mathcal{O}_{\mathcal{I}}$ binds always the least functional variable of relations. In our example, it is more reasonable to predict the place of residence of a person, that all the people that live in a place (Section 4.2). Once the arguments are fixed, the atom is appended to the rule, and the user can choose the next atom. In the example, the user would select the relation $isMarriedTo$, followed by the arguments a, b, and then in a new atom the relation $livesIn$ (Figure 1). The user can also, at any point, decide to backtrack and to remove the last atom from the rule in order to explore a different path in the search space.

At each step in the process, the user has complete freedom to choose his options. However, the system also shows the metrics and pruning strategies that AMIE would apply. Our demo shows for every possible choice of atoms the support, the head coverage, the standard confidence, and the PCA confidence that this atom would achieve. Thus, the user can see which choice achieves the best metrics, and why – much like AMIE does during the mining process. For illustration, we also show positive examples, negative examples, and negative examples under the PCA for the new rule.

5.2 Implementation

Our demo is implemented as a client-server application that lets the user drive the AMIE algorithm step by step. We use the YAGO ontology [19] as KB. In order to guarantee reasonable response times in the interactive client-server setting, we created a sample of YAGO following the same procedure as in [10]: We took 10K random seed entities, and collected all the facts within a range of 3 hops from the seed entities, producing a sample of 35K facts. The server consists of a Java servlet that serves as interface to the AMIE codebase, namely the mining operators described in Section 4.3 and the in-memory triple store. The KB is loaded into memory only once when the servlet is initialized in the servlets container. The client side is a lightweight user-interface written in Javascript and HTML.

For the AMIE algorithm, it does not matter whether an unknown person or President Obama is a counter-example for a rule. For the user, in contrast, it is more illustrative to show prominent entities rather than unknown ones. Hence, we computed a relevance score for each entity e as:

$$relevance(e) := log(wikilength(e)) \times (incoming(e) + 1)$$

Here, $wikilength(e)$ is the length of the Wikipedia article of the entity (in bytes), and $incoming(e)$ is the number of Wikipedia articles linking to the article of e. Both numbers can be easily obtained from YAGO. We add 1 in the second term to guarantee that the score is a positive number. The relevance of a fact $r(x, y)$ is defined as the sum of the relevance scores of its arguments. This score is used to rank facts when displaying examples for rules, so that facts about prominent entities are preferred.

6. CONCLUSION

Rule Mining is a fascinating process, in which nuggets of insightful correlations are mined on millions of facts. Our demo allows the user to join AMIE in this process on a sample KB, and to discover how the search space can be cut down to a tractable size. Our demo is fully implemented, and available online at `http://luisgalarraga.de/amie-demo/`. Through the interaction with the demo participants, we hope not just to shed light on the inner workings of AMIE, but also to develop ideas for new pruning strategies together with the participants.

7. ACKNOWLEDGMENTS

This work is supported by the Chair "Machine Learning for Big Data" of Télécom ParisTech.

8. REFERENCES

[1] Y. Amsterdamer, Y. Grossman, T. Milo, and P. Senellart. Crowd miner: Mining association rules from the crowd. In *Proc. VLDB*, Riva del Garda, Italy, 2013. Demo paper.

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A nucleus for a web of open data. In *ISWC*, 2007.

[3] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. H. Jr., and T. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, 2010.

[4] L. Dehaspe and H. Toironen. Discovery of relational association rules. In *Relational Data Mining*. Springer-Verlag New York, Inc., 2000.

[5] L. Dehaspe and H. Toivonen. Discovery of frequent DATALOG patterns. *Data Min. Knowl. Discov.*, 3(1), Mar. 1999.

[6] L. Del Corro and R. Gemulla. Clausie: clause-based open information extraction. In *WWW*, 2013.

[7] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *KDD*, 2014.

[8] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam. Open Information Extraction: the Second Generation. In *IJCAI*, 2011.

[9] L. Galárraga, G. Heitz, K. Murphy, and F. Suchanek. Canonicalizing Open Knowledge Bases. In *CIKM*, 2014.

[10] L. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek. AMIE: Association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*, 2013.

[11] L. A. Galárraga, N. Preda, and F. M. Suchanek. Mining rules to align knowledge bases. In *AKBC*, 2013.

[12] B. Goethals and J. Van den Bussche. Relational Association Rules: Getting WARMER. In *Pattern Detection and Discovery*, volume 2447. Springer Berlin / Heidelberg, 2002.

[13] X. Lin, A. Mukherji, E. A. Rundensteiner, and M. O. Ward. Spire: Supporting parameter-driven interactive rule mining and exploration. *Proc. VLDB Endow.*, 2014.

[14] S. Muggleton. Inverse entailment and progol. *New Generation Comput.*, 13(3&4), 1995.

[15] S. Muggleton. Learning from positive data. In *ILP*. Springer-Verlag, 1997.

[16] V. Nebot and R. Berlanga. Finding association rules in semantic web data. *Knowl.-Based Syst.*, 25(1), 2012.

[17] S. Schoenmackers, O. Etzioni, D. S. Weld, and J. Davis. Learning first-order Horn clauses from web text. In *EMNLP*, 2010.

[18] F. M. Suchanek, S. Abiteboul, and P. Senellart. Paris: Probabilistic alignment of relations, instances, and schema. *PVLDB*, 5(3), 2011.

[19] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A Core of Semantic Knowledge. In *WWW*, 2007.

[20] J. Voelker and M. Niepert. Statistical schema induction. In *ESWC*, 2011.

[21] Q. Zeng, J. Patel, and D. Page. QuickFOIL: Scalable Inductive Logic Programming. In *VLDB*, 2014.