Préface of Proceedings of the conference BDA'2021

Esther Pacitti, Zoltan Miklos

BDA 2021

37ème Conférence sur la Gestion de Données Principes Technologies et Applications

Esther Pacitti[1] and Zoltan Miklos[2]

[1]LIRMM, Université de Montpellier. esther.pacitti@lirmm.fr
[2]IRISA, Université de Rennes 1. zoltan.miklos@irisa.fr

Actes de la conférence BDA 2021

Site web de la conférence : `https://bda2021.inria.fr/`

Table des matières

1 Message du Président et des organisateurs

La conférence BDA : Gestion de Données – Principes, Technologies et Applications est le rendez-vous annuel incontournable de la communauté en gestion de données en France. Sa 37ème édition s'est tenue du 25 au 28 octobre 2021. A cause de la crise sanitaire liée au COVID-19, elle s'est déroulée en ligne du 27 au 27 octobre et en présentiel le 28 octobre à Paris dans les locaux de ENS Paris.

Poursuivant la tradition des rencontres annuelles de la communauté de gestion de données francophone, BDA 2021 a invité les acteurs académiques et industriels de la recherche en gestion de données à soumettre leurs travaux récents afin de présenter les défis et les avancées scientifiques dans ce domaine extrêmement dynamique.

Avec plus de 200 participants (dont 80 en présentiel), BDA atteste une nouvelle fois du dynamisme de notre communauté et de l'importance de la gestion de données. La recherche en gestion de données n'a jamais été aussi active, variée et ouverte sur d'autres champs de l'informatique. L'omniprésence des données massives change en profondeur la manière dont les différentes phases du processus d'acquisition et de valorisation des données sont mises en œuvre. Ces données sont volumineuses, hétérogènes, incomplètes, imprécises, produites dynamiquement et avec divers degrés de structuration. L'exploitation de ces données par des applications issues de domaines scientifiques et métiers très variés pose de nombreux défis pour la communauté de recherche en informatique tant sur le plan fondamental qu'appliqué. Cette évolution s'inscrit aujourd'hui dans le contexte de la science des données, avec le cercle vertueux entre données massives et apprentissage automatique, apportant de nouveaux défis à notre communauté.

Le programme scientifique final a comporté 33 articles de recherche, dont 25 longs et 8 courts, 6 démonstrations et 8 articles de doctorants. Il a été complété par trois conférences invitées sur des sujets d'actualité. Un objectif important de BDA est de donner la possibilité aux chercheurs, et surtout aux doctorants, de présenter leurs travaux à la communauté, ce qui inclut également des travaux récents déjà publiés dans d'autres conférences. Les actes de la conférence proposent ainsi des résumés de toutes les contributions publiées et non-publiées et sont complétés par une édition spéciale du journal Transactions on Large-Scale Data and Knowledge-Centered Systems (TLDKS), qui rassemble les versions étendues de quelques articles sélectionnés. Comme les années précédentes, BDA 2011 a proposé également des prix pour des contributions exceptionnelles, récompensant quatre articles de recherche, une démonstration, ainsi qu'une thèse de la communauté indiquée à la fin de ces actes.

Nous tenons à remercier tous les auteurs pour la qualité de leurs contributions et présentations, les membres de l'équipe d'organisation, les membres des comités de programme et du prix de thèse. Nous remercions en particulier les conférenciers invités Anastasia Ailamaki, Georgia Koutrika et Benjamin Nguyen, pour avoir illuminé la conférence par l'excellence de leurs présentations. Nous sommes également reconnaissants envers nos soutiens qui ont permis d'organiser cette manifestation, en particulière ENS Paris, l'équipe Valda (DI ENS Paris, PSL), l'équipe BD, LIP6 Sorbonne Universités, l'équipe DRUID, IRISA Rennes, et INSA Valor. Enfin, nos remerciements vont aux nombreux participants qui ont fait vivre cette belle édition 2021.

Philipp Rigaux, Président des journées
Zoltan Miklos, Président du Comité d'Organisation
Esther Pacitti, Président du Comité de Programme

2 Présidence & comités BDA 2021

2.1 Présidente des journées

— Philippe Rigaux, CNAM Paris

2.2 Comité d'organisation

— Zoltan Miklos, IRISA, Université de Rennes 1 (président)
— Hélène Jaudoin, IRISA, Université de Rennes 1
— François Goasdoué, IRISA, Université de Rennes 1
— Virginie Thion, IRISA, Université de Rennes 1
— Laurent d'Orazio, IRISA, Université de Rennes 1
— Mickaël Foursov, IRISA, Université de Rennes 1
— Annie Foret, IRISA, Université de Rennes 1
— Sébastien Ferré, IRISA, Université de Rennes 1
— Michael Thomazo, INRIA, ENS PSL
— Camille Bourgaux, CNRS, ENS PSL
— Bernd Amann, LIP6, Sorbonne Université

2.3 Comité de programme

— Esther Pacitti, LIRMM, Université de Montpellier (présidente)
— Bernd Amann, LIP6, Sorbonne Université
— Omar Boucelma, LSIS, Université Aix-Marseille
— Amel Bouzeghoub, Télécom SudParis
— Sylvie Cazalens, LIRIS, INSA de Lyon
— Sarah Cohen-Boulakia, LRI, Université Paris-Sud
— Alexandru Constan, INRIA
— Helena Galhardas, Université de Lisbonne
— François Goasdoué, IRISA, Université de Rennes 1
— Daniela Grigory, LAMSADE, Université Paris-Dauphine
— David Gross-Amblard, IRISA, Université Rennes 1
— Mirian Halfeld Ferrari, LIFO, Université d'Orléans
— Zoubida Kedad, DAVID, Université de Versailles-Saint-Quentin
— Anne Laurent, LIRMM, Université de Montpellier
— Ioana Manolescu, INRIA, Institut Polytechnique de Paris
— Riad Mokadem, IRIT, Université Paul Sabatier
— Pascal Molli, LS2N, Université de Nantes
— Cedric Mouza, CNAM
— Hubert Naake LIP6, Sorbonne Université
— Paolo Papotti, EUROCOM
— Jorge-Arnulfo Quiané-Ruiz, TU Berlin
— Claudia Ronconcio, LIG, Université de Grenoble
— Marie-Christine Rousset, LIG, Université de Grenoble Alpes
— Patricia Serrano, LS2N, Université de Nantes

— Maximilien Servajean, LIRMM, Université de Montpellier
— Dennis Shasha, New York University
— Hala Skaf-Molli, LS2N, University of Nantes
— Farouk Toumani, LIMOS, Université Clermont Auvergne
— Patrick Valduriez, INRIA, Université de Montpellier
— Genoveva Vargas, LIRIS, INSA de Lyon
— Dan Vodislav, ETIS, University of Cergy-Pontoise
— Karine Zeitouni, DAVID, Université de Versailles-Saint-Quentin

2.4 Comité de démonstration

— Reza Akbarinia, INRIA (président)
— Tristan Allard, IRISA, Université de Rennes
— Oana Balalau, Inria and École Polytechnique
— Laure Berti-Equille, IRD
— Dario Colazzo, LAMSADE, Université Paris-Dauphine
— Camélia Constantin, LIP6, Sorbonne Université
— Amelie Gheerbrant, IRIF, CNRS, Université de Paris
— Hélène Jaudoin, IRISA, Université de Rennes 1
— Florent Masseglia, INRIA
— Nada Mimouni, Le CNAM
— Tanmoy Mondal, IMT Atlantique
— Marian Scuturici, LIRIS, INSA de Lyon
— Nicolas Travers, Léonard de Vinci Pôle Universitaire
— Katerina Tzompanaki, ETIS, Université de Cergy-Pontoise
— Chao Zhang, LIRIS, Université Lyon 1

2.5 Comité du prix de thèse

— Jean-Marc Petit, LIRIS, INSA de Lyon (président)
— Laure Berti-Equille, IRD, Montpellier
— Pierre Bourhis, Cristal, CNRS, Lille
— Daniela Grigori, Lamsade, Paris Dauphine
— Nabil Layaïda, Tyrex, INRIA, Grenoble
— Ioana Manolescu, Cedar, INRIA, Paris Saclay
— Sofian Maabout, Labri, Bordeaux
— Florent Masseglia, Zenith, INRIA, Montpellier

3 Conférences invitées

3.1 Nothing is for granted : Making wise decisions using real-time intelligence
Anastasia Ailamaki

In today's ever-growing demand for fast, data-driven decisions, heterogeneity severely undermines performance and fragments efforts for building unified data exploration tools. The variety in data formats and workloads forces data pipelines to be manually split across a variety of task-specialized systems and combined through expensive ETL and orchestration processes, or to adapt both the data and the workloads to match the requirements of a single-system, sacrificing expressiveness and structural information. Furthermore, the ever-increasing hardware heterogeneity causes task-based specialization of the tools to specific hardware such as CPUs or GPUs, forcing a trade-off : designing optimized hardware often means wasting accelerator-level parallelism (ALP) opportunities or tolerating slow and unnecessary communication between devices. In general, data processing is adapted to the pre-determined data processing system architecture, losing valuable information in the translation.

Real-time intelligence means to make all decisions during execution, when all relevant information is available for optimal utilisation of resources, while it also learns and extracts information about the query requests, instead of depending on pre-determined workload expectations. I will show how designing top-down the system architecture to allow a data- and workload-driven just-in-time specialization enables fast query execution over unprepared, potentially dirty data without time consuming preparation, as well as efficient orchestration and utilization of heterogeneous hardware devices.

Anastasia Ailamaki is a Professor of Computer and Communication Sciences at the École Polytechnique Fédérale de Lausanne (EPFL) in Switzerland and the co-founder of RAW Labs SA, a Swiss company developing real-time analytics infrastructures for heterogeneous big data from multiple sources. She earned a Ph.D. in Computer Science from the University of Wisconsin-Madison in 2000. She received the 2019 ACM SIGMOD Edgar F. Codd Innovations and the 2020 VLDB Women in Database Research Award. She is also the recipient of an ERC Consolidator Award (2013), the Finmeccanica endowed chair from the Computer Science Department at Carnegie Mellon (2007), a European Young Investigator Award from the European Science Foundation (2007), an Alfred P. Sloan Research Fellowship (2005), an NSF CAREER award (2002), and ten best-paper awards in database, storage, and computer architecture conferences. She is an ACM fellow, an IEEE fellow, the Laureate for the 2018 Nemitsas Prize in Computer Science, and an elected member of the Swiss, the Belgian, the Greek, and the Cypriot National Research Councils. She is a member of the Academia Europaea and of the World Economic Forum Expert Network.

3.2 The Rise of Intelligent Data Assistants : Democratizing Data Access
Georgia Koutrika

For many, data is considered the 21st century's most valuable commodity growing at an exponential rate – but is it for everyone ? Analysts exploring data sets for insight, scientists looking for patterns, and consumers looking for information are just a few examples of user groups that need to access and dig into data. However, existing data exploration tools are falling behind in bridging the chasm between data and users, making data exploration intended only for the few. In this talk, we will discuss about what it takes to bridge this chasm and the new generation of intelligent data exploration tools that are emerging at the intersection of data management, natural language processing, machine learning and visualization. The talk will end with a summary of open questions on intelligent data assistants.

Georgia Koutrika is a Research Director at Athena Research Center in Greece. She has more than 15 years of experience in multiple roles at HP Labs, IBM Almaden, and Stanford. She has received a PhD and a diploma in Computer Science from the Department of Informatics and Telecommunications, University of Athens, Greece. Her work focuses on data exploration, recommendations, and data analytics, and has been incorporated in commercial products, described in 14 granted patents

and 26 patent applications in the US and worldwide, and published in more than 90 papers in top-tier conferences and journals. Georgia is an ACM Senior Member, IEEE Senior Member, and ACM Distinguished Speaker. Her recent academic activities include : Editor-in-chief for VLDB Journal, PC co-chair for VLDB 2023, Co-EiC of Proceedings of the VLDB (PVLDB) Vol 16, and associate editor for TKDE, SIGMOD2022 and VLDB2022.

3.3 Privacy Preserving Contact Tracing : a case of privacy driven data analysis *Benjamin Nguyen*

Classical contact-tracing is a technique during an epidemy to slow down contamination. Classical contact tracing is organized by medical personnel during an (intrusive) interview with a patient suffering from the disease, in order to trace back all the other people whom the patient was in contact with, in order to informe them of the possibility of contamination.

During the Covid19 pandemic, contact tracing has been pushed to a new level, via automation. However, due to the sensitive nature of the information that is manipulated in order to achieve good tracing, building a privacy preserving solution, while still maintaining the quality of the results proved to be a compelling research challenge. During this presentation, we will present the data management and privacy issues around privacy preserving contact tracing, and discuss in more detail the two main solutions deployed in Europe, the French lead ROBERT (aka StopCovid/TousAntiCovid) proposal, and the Swiss lead DP3T, later adapted by Apple and Google.

Benjamin Nguyen graduated from ENS Cachan in 2000. He obtained his Ph.D from University of Paris Sud in 2003 on data warehousing, and his HDR from University of Versailles in 2013 on privacy preserving data mangement. Since 2014, he holds a professor position in the Systems and Data Security team of the Laboratoire d'Informatique Fondamentale d'Orléans (LIFO), at INSA Centre Val de Loire and is an external collaborator of the Inria Private and Trusted Cloud (PETRUS) team. His current research topics cover anonymization techniques, models and methods to represent, quantify and enforce privacy models (such as logics, secure hardware and cryptography), and the design and implementation of large scale privacy-by-design information management systems. Pr. Nguyen is head of LIFO since 2016, and co-chair of the Privacy Working group of the CNRS research group on Computer Security since 2016.

4 Résumés des articles long

On Predictive Explanation of Data Anomalies

Nikolaos Myrtakis
myrtakis@csd.uoc.gr
University of Crete
Heraklion, Greece

Ioannis Tsamardinos
tsamard.it@gmail.com
University of Crete
Heraklion, Greece

Vassilis Christophides
vassilis.christophides@ensea.fr
ENSEA
Cergy, France

KEYWORDS

Anomaly Explanation, Predictive Explanation, Anomaly Interpretation, Explainable AI

1 INTRODUCTION

Detection of "anomalous" samples (records, instances), called *anomaly detection*, is an important problem in machine learning. It is conceptually related to outlier and novelty detection in several application settings. The anomalous samples may indicate mislabelled data, catastrophic measurements or data entry errors, bugs in data wrangling and preprocessing software, or other interesting phenomena.

Numerous **unsupervised** algorithms (e.g., IF [4], LOF [1], LODA [7]) to detect anomalies (hereafter **detectors**) have been proposed. The most advanced ones detect anomalies in a multi-dimensional fashion, simultaneously considering all feature values. Unfortunately, detectors, in general, do not explain why a sample was considered as abnormal, leaving human analysts with no guidance about their root causes, insight to take corrective actions, or remedy their effect.

Several methods for **explaining anomalies** have been proposed, hereafter **explainers**. *The explanations often take the form of a subset of features* called a **subspace** in the literature. The idea is that *by examining only the explaining features suffices to determine whether the sample is an anomaly or not according to the detector.*

Existing methods can be categorized to those that provide **local explanations** (point-based) that pertain to a single sample, or **global explanations** (a.k.a. set-based) to simultaneously explain all training samples. The latter is important in order to reduce the burden of human analysts to inspect possibly different explanations for each anomaly. We should stress that global explanation is different from clustering as the former's objective is to provide a subspace segregating the anomalous from normal samples. Explainers may be **specific** to a detection algorithm or detector-**agnostic**, hence applicable post-hoc to any detection algorithm. As reported by several independent experimental studies, e.g. [2], there is no detector outperforming all others on all possible datasets. Hence, researchers cannot just design a specific explainer for the optimal detector; it may thus be preferable to design optimal agnostic explainers. Explainers may also be categorized as **descriptive** in the sense that they explain the samples used to train the detector. Explainers that return explanations that generalize to unseen data are **predictive** ones. The importance of predictive explanations has been recognised in Explainable AI to avoid recomputing explanations on every new batch of data.

Figure 1 illustrates how predictive explanations can be used in data validation pipelines monitoring the data fed to downstream ML models. Given that in real application settings it is difficult or even impossible to label data as anomalous or normal [2], unsupervised detectors are initially used to spot anomalies. Then, a predictive anomaly explainer could be used by human analysts to reveal the root causes of the detected anomalies and decide subsequent corrective actions. It is essentially a surrogate model, trained with a small subset of the original features that serve as explaining feature subspace. Depending on the quality of the approximation of the decision boundary of an unsupervised detector, the surrogate model can be also used to detect anomalies in fresh data, i.e., new batches of data, by completely bypassing the need to rerun the detector.

In this paper, we propose **a novel method to produce global, predictive explanations** called **PROTEUS**[1]. PROTEUS is *detector agnostic*, and can be used to approximate the decision boundary of any detector. We should stress that prior work on detector agnostic explainers like CA-Lasso [6] and SHAP [5] but also detector specific explainers like LODA [7] produce explanations that are only **local and descriptive**.

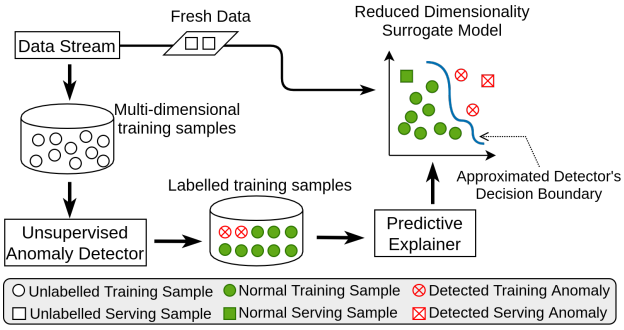


Figure 1: Predictive Anomaly Explanation Pipeline

PROTEUS essentially constructs a *reduced-dimensionality, surrogate model* that approximates the behavior of a detector with fewer features. Since the detector is labelling the samples as anomalies or not, the problem of finding such a model reduces to a *supervised predictive modeling with feature selection* problem. In order for the surrogate model to also explain unseen samples, it has to approximate the detector's decision boundary and not simply interpolate the anomalies (overfit) in the training data. To this respect, *the*

[1]Proteus or Πρωτεύς in Greek, means 'first' and is a minor sea God and son of Poseidon.

quality of approximation should be estimated using out-of-sample performance estimation protocols like K-fold cross validation (CV). To build the model, any combination of feature selection algorithm with a classifier could be employed. However, ideally one should optimize the combination of algorithms and their hyper-parameter values to achieve the best approximation with the samples at hand.

The above requirements for tuning and estimating generalization performance of predictive models are nowadays addressed in the automated machine learning (**AutoML**) systems [3]. In this respect, **producing predictive anomaly explanations can be solved as an AutoML problem**. Unfortunately, the majority of existing tools such as auto-sklearn do not perform feature selection. In addition, they do not exploit the fact that the data can be augmented with new samples (pseudo-samples) that can be labelled by the detector, to improve performance. Finally, their performance estimates are often overestimated[2], particularly for imbalanced datasets. To address the above issues, PROTEUS makes the following contributions:

(1) We introduce a novel AutoML engine[3] specifically designed to support feature selection and classification on imbalanced datasets. Unlike existing explainers, PROTEUS outputs not only a *small-sized feature subset serving as explanation* but also a *surrogate model fitted on this subset* to explain unseen samples, as well as a *reliable out-of-sample (predictive) performance estimation.*

(2) To produce such output, PROTEUS AutoML relies on *advanced design choices*, such as supervised oversampling, group-based stratification, and a special variant of Cross-Validation with Bootstrap Bias Correction (**BBC**) [8].

(3) Thorough computational experiments we show the efficacy and robustness of PROTEUS in synthetic and real datasets of increasing dimensionality. Last but not least, our experiments show that PROTEUS approximates accurately the performance of a specific explainer (LODA) in a detector-agnostic fashion.

(4) We formally define descriptive and predictive explanations, originally introduced in our work.

(5) We assess the merit of the idea to use PROTEUS to correct the decisions of the unsupervised anomaly detectors. Specifically, we study the disagreements of classification to anomalies between the surrogate PROTEUS model and the detector. We show that PROTEUS can often correct the false positives of false negatives as identified by the detector.

(6) We propose a new visualization method for presenting the global explanations found by PROTEUS as spider charts. The visualizations provide insight regarding the combination of feature values that lead to calling a sample as anomalous or not.

(7) We position PROTEUS w.r.t. various categories of related work on explaining anomalies in unsupervised and supervised settings.

The full text may be found at https://arxiv.org/abs/2110.09467.

REFERENCES

[1] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *SIGMOD '00.*

[2] Markus Goldstein and Seiichi Uchida. 2016. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLoS One* (2016).

[3] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren (Eds.). 2018. *Automated Machine Learning: Methods, Systems, Challenges.*

[4] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation Forest. In *ICDM.*

[5] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *NeurIPS.* 4765–4774.

[6] Barbora Micenková, Raymond T. Ng, Xuan-Hong Dang, and Ira Assent. 2013. Explaining Outliers by Subspace Separability. In *ICDM.* 518–527.

[7] Tomás Pevný. 2015. Loda: Lightweight on-line detector of anomalies. *Machine Learning* 102 (2015), 275–304.

[8] Ioannis Tsamardinos, Elissavet Greasidou, and Giorgos Borboudakis. 2018. Bootstrapping the out-of-sample predictions for efficient and accurate cross-validation. *Mach. Learn.* 107, 12 (2018), 1895–1922.

[2] https://www.kdnuggets.com/2020/12/trust-automl.html

[3] https://git.io/JtCwU

Efficient Incremental Computation of Aggregations over Sliding Windows

Chao Zhang
LIMOS, CNRS, University of Clermont Auvergne
France
chao.zhang@uca.fr

Reza Akbarinia
LIRMM, INRIA, University of Montpellier
France
reza.akbarinia@inria.fr

Farouk Toumani
LIMOS, CNRS, University of Clermont Auvergne
France
farouk.toumani@uca.fr

ABSTRACT

Computing aggregation over sliding windows, *i.e.*, finite subsets of an unbounded stream, is a core operation in streaming analytics. We propose PBA (Parallel Boundary Aggregator) a novel parallel algorithm that groups continuous slices of streaming values into chunks and exploits two buffers, cumulative slice aggregations and left cumulative slice aggregations, to compute sliding window aggregations efficiently. PBA runs in $O(1)$ time, performing at most 3 merging operations per slide while consuming $O(n)$ space for windows with n partial aggregations. Our empirical experiments demonstrate that PBA can improve throughput up to 4× while reducing latency, compared to state-of-the-art algorithms.

CCS CONCEPTS

• **Information systems** → **Data streaming**.

KEYWORDS

Data Stream, Streaming Algorithm, Sliding Window Aggregation

1 INTRODUCTION

Nowadays, we are witnessing the production of large volumes of continuous or real-time data in many application domains like traffic monitoring, medical monitoring, social networks, weather forecasting, network monitoring, etc. For example, every day around one trillion messages are processed through Uber data analytics infrastructure[1] while more than 500 million tweets are posted on Twitter [6]. Efficient streaming algorithms are needed for analyzing data streams in such applications. In particular, aggregations [12], having the inherent property of summarizing information from data, constitute a fundamental operator to compute real-time statistics in this context. In the streaming setting, aggregations are typically computed over finite subsets of a stream, called *windows*. In particular, Sliding-Window Aggregation (SWAG) [10, 11, 14, 16] continuously computes a summary of the most recent data items in a given range r (aka window size) and using a given slide s. If the range and slide parameters are given in time units (*e.g.*, seconds), then the sliding window is *time-based*, otherwise, *i.e*, if these parameters are given as the number of values, it is *count-based*. Fig. 1 presents an example of computing *sum* over the count-based sliding window with a range of 10 values and a slide of 2 values.

Stream processing systems (SPSs) [1, 4, 9, 17, 19] are ubiquitous for analyzing continuously unbounded data. However, one of the challenges faced by SPSs is to efficiently compute aggregations over sliding windows. This requires the ability of such systems to incrementally aggregate moving data, *i.e.*, inserting new data items and evicting old data items when a window is sliding without recomputing the aggregation from scratch. High throughput and low latency are essential requirements as SPSs are typically designed for real-time applications [7].

Two orthogonal techniques have been proposed to meet such requirements: *slicing* (aka partial aggregation) [5, 8, 10, 18], and *merging* (aka incremental final aggregation) [13–16]. Slicing techniques focus on slicing the windows into smaller subsets, called *slices*, and then computing partial aggregation over slices, called *slice aggregations*. The benefit of the slicing technique is that slice aggregations (*i.e.*, partial aggregations over slices) can be shared by different window instances. For example, in Fig. 1, a slice aggregation over a slice of 2 values can be shared by 5 windows. On the basis of a slicing technique, final aggregations over sliding windows are computed by merging slice aggregations, *e.g.*, in Fig. 1, SWAG is computed by merging 5 slice aggregations. During the merging phase, each insertion or eviction is processed over a slice aggregation rather than a value. In modern SPSs, practical sliding windows can be very large [16], thereby making the merging phase non-trivial. To efficiently merge slice aggregation, *incremental computation* is necessary because an approach that recalculate slice aggregations from scratch (hereafter called Recal) is very inefficient [10, 11, 14, 16].

The difficulty of incremental merging depends on the considered class of aggregations: *invertible* or *non-invertible*. An aggregation is invertible if the merging function for its partial aggregations has an inverse function. For example, *sum* is clearly invertible, because it has the arithmetical subtraction as the inverse function. Similarly, *avg* and *std* are invertible because they use *addition* as merging function. Partial aggregations of invertible aggregations can be efficiently merged using the *subtract-on-evict* algorithm [7, 15], *i.e.*, maintaining a running *sum* over a sliding window by subtracting evicted values and adding inserted values. However, this is not the case for non-invertible aggregations, *e.g.*, *max*, *min*, and *bloom filter*, where the *subtract-on-evict* algorithm cannot be applied at the merging phase. Incremental merging of continuous data in the context of non-invertible aggregations is challenging and requires more sophisticated algorithms (*e.g.*, see [3, 13–16]).

[1] AthenaX: SQL-based streaming analytics platform, https://eng.uber.com/athenax

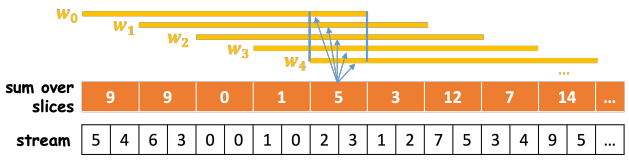


Figure 1: Example of computing *sum* over a count-based sliding window with a range of 10 values and a slide of 2 values. SWAG computation consists of two phases: (1) computing the partial aggregations *sum* over 2 values; (2) merging 5 partial aggregations.

This paper focuses on merging slice aggregations for computing non-invertible aggregations over FIFO sliding windows with an arbitrary range and an arbitrary slide. We propose PBA (Parallel Boundary Aggregator), a novel algorithm that computes incremental aggregations in parallel. PBA groups continuous slices into *chunks*, and maintains two buffers for each chunk containing, respectively, the *cumulative slice aggregations* (denoted as *csa*) and the *left cumulative slice aggregations* (denoted as *lcs*) of the chunk's slices. Using such a model, SWAGs can be computed in constant time bounded by 3 for both amortized and worst-case time. Interestingly, the required *csa* and *lcs* for each window aggregation are completely independent from each other, so the incremental computation of *csa* and *lcs* can be achieved in parallel. These salient features put PBA ahead of state-of-the-art algorithms in terms of throughput and latency.

In this paper, we make the following main contributions:

- We propose a novel algorithm, PBA, to efficiently compute SWAGs. PBA uses chunks to divide final aggregations of window instances into sub-aggregations that are elements of *csa* and *lcs* buffers. These two buffers can be computed incrementally and in parallel.
- We analyze the latency caused by SWAG computations with different chunk sizes in PBA, and propose an approach to optimize the chunk size leading to the minimum latency.
- We conduct extensive empirical experiments, using both synthetic and real-world datasets. Our experiments show that PBA behaves very well for average and large sliding windows (*e.g.*, with sizes higher than 1024 values), improving throughput up to 4× against state-of-the-art algorithms while reducing latency. For small-size windows, the results show the superiority of the non-parallel version of PBA (denoted as SBA) that outperforms other algorithms in terms of throughput.
- To show the benefit of our approach in modern stream-processing frameworks, we implemented PBA on top of Apache Flink [2, 4], called FPBA. Our empirical evaluation shows that FPBA scales well as increasing the parallelism of Flink in both local and cluster modes.

All technical details are included in the full version of this paper [20].

2 ACKNOWLEDGMENTS

This research was financed by the French government IDEX-ISITE initiative 16-IDEX-0001 (CAP 20-25).

REFERENCES

[1] Tyler Akidau, Alex Balikov, Kaya Bekiroğlu, Slava Chernyak, Josh Haberman, Reuven Lax, Sam McVeety, Daniel Mills, Paul Nordstrom, and Sam Whittle. 2013. MillWheel: Fault-Tolerant Stream Processing at Internet Scale. *Proc. VLDB Endow.* (2013).
[2] Apache Flink. 2019. https://flink.apache.org.
[3] Arvind Arasu and Jennifer Widom. 2004. Resource Sharing in Continuous Sliding-Window Aggregates. In *VLDB*.
[4] Paris Carbone, Asterios Katsifodimos, Stephan Ewen, Volker Markl, Seif Haridi, and Kostas Tzoumas. 2015. Apache Flink: Stream and Batch Processing in a Single Engine. *IEEE Data Eng. Bull.* (2015).
[5] Paris Carbone, Jonas Traub, Asterios Katsifodimos, Seif Haridi, and Volker Markl. 2016. Cutty: Aggregate Sharing for User-Defined Windows. In *CIKM '16*.
[6] Vibhuti Gupta and Rattikorn Hewett. 2020. Real-Time Tweet Analytics Using Hybrid Hashtags on Twitter Big Data Streams. *Inf.* (2020).
[7] Martin Hirzel, Scott Schneider, and Kanat Tangwongsan. 2017. Sliding-Window Aggregation Algorithms: Tutorial. In *DEBS*.
[8] Sailesh Krishnamurthy, Chung Wu, and Michael Franklin. 2006. On-the-Fly Sharing for Streamed Aggregation. In *SIGMOD*.
[9] Sanjeev Kulkarni, Nikunj Bhagat, Maosong Fu, Vikas Kedigehalli, Christopher Kellogg, Sailesh Mittal, Jignesh M. Patel, Karthik Ramasamy, and Siddarth Taneja. 2015. Twitter Heron: Stream Processing at Scale. In *SIGMOD*.
[10] Jin Li, David Maier, Kristin Tufte, Vassilis Papadimos, and Peter A. Tucker. 2005. No Pane, No Gain: Efficient Evaluation of Sliding-Window Aggregates over Data Streams. *SIGMOD Rec.* (2005).
[11] Jin Li, David Maier, Kristin Tufte, Vassilis Papadimos, and Peter A. Tucker. 2005. Semantics and Evaluation Techniques for Window Aggregates in Data Streams. In *SIGMOD*.
[12] Radko Mesiar Michel Grabisch, Jean-Luc Marichal and Endre Pap. 2009. *Aggregation Functions.* Cambridge University Press.
[13] Anatoli U. Shein, Panos K. Chrysanthis, and Alexandros Labrinidis. 2017. FlatFIT: Accelerated Incremental Sliding-Window Aggregation For Real-Time Analytics. In *SSDBM*.
[14] Anatoli U. Shein, Panos K. Chrysanthis, and Alexandros Labrinidis. 2018. SlickDeque: High Throughput and Low Latency Incremental Sliding-Window Aggregation. In *EDBT*.
[15] Kanat Tangwongsan, Martin Hirzel, and Scott Schneider. 2017. Low-Latency Sliding-Window Aggregation in Worst-Case Constant Time. In *DEBS*.
[16] Kanat Tangwongsan, Martin Hirzel, Scott Schneider, and Kun-Lung Wu. 2015. General Incremental Sliding-Window Aggregation. *PVLDB* (2015).
[17] Ankit Toshniwal, Siddarth Taneja, Amit Shukla, Karthik Ramasamy, Jignesh M. Patel, Sanjeev Kulkarni, Jason Jackson, Krishna Gade, Maosong Fu, Jake Donham, Nikunj Bhagat, Sailesh Mittal, and Dmitriy Ryaboy. 2014. Storm@twitter. In *SIGMOD*.
[18] Jonas Traub, Philipp M. Grulich, Alejandro Rodriguez Cuellar, Sebastian Breß, Asterios Katsifodimos, Tilmann Rabl, and Volker Markl. 2019. Efficient Window Aggregation with General Stream Slicing. In *EDBT*.
[19] Matei Zaharia, Tathagata Das, Haoyuan Li, Timothy Hunter, Scott Shenker, and Ion Stoica. 2013. Discretized Streams: Fault-Tolerant Streaming Computation at Scale. In *SOSP*.
[20] Chao Zhang, Reza Akbarinia, and Farouk Toumani. 2021. Efficient Incremental Computation of Aggregations over Sliding Windows. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 2136–2144. https://doi.org/10.1145/3447548.3467360

Incremental Schema Discovery at Scale for RDF Data

Redouane Bouhamoum, Zoubida Kedad and Stéphane Lopes
DAVID lab. - University of Versailles Saint-Quentin-en-Yvelines
Versailles, France
firstName.lastName@uvsq.fr

ABSTRACT

The lack of a descriptive schema for an RDF dataset has motivated several research works addressing the problem of automatic schema discovery. The goal of these approaches is to generate the schema of an RDF dataset by analysing its instances. However, as new instances are added, the schema may become inconsistent with the dataset. In this paper, we propose an incremental schema discovery approach for massive RDF datasets. It is based on a scalable and incremental density-based clustering algorithm which propagates the changes occurring in the dataset into the clusters corresponding to the classes of the schema. Our approach is implemented using big data technology. We present some experiments which demonstrate the efficiency of our proposal.

CCS CONCEPTS

• **Information systems** → *Semantic web description languages*; **Resource Description Framework (RDF)**; **Clustering**; • **Computing methodologies** → Parallel algorithms.

KEYWORDS

Incremental Schema Discovery, RDF Data, Big Data, Clustering.

1 INTRODUCTION

The Web of data represents a huge information space consisting of an increasing number of interlinked datasets described using the Resource Description Framework (RDF). These datasets contain both the data and the schema describing the data. However, the schema may be incomplete or missing which limits the use of RDF data. Indeed, it is not obvious to explore a dataset without a schema describing its contents.

We have proposed in previous works a schema discovery approach suitable for very large datasets, which relies on a scalable density-based clustering algorithm [1]. However, RDF datasets are subject to frequent evolutions over time, and new instances may be inserted. Due to such evolution, the ability to perform incremental updates on the schema has emerged as a new challenge.

In this work, we propose an incremental schema discovery approach for large RDF datasets. Our contribution is an incremental density-based clustering algorithm for building and updating the clusters that represent the classes of the schema. Our algorithm incrementally updates the classes to keep them consistent with the evolution of the data and ensures providing the same result as a sequential clustering algorithm. In addition, the incremental clustering process is parallelized and implemented using Spark to be efficient on large datasets.

2 AN INCREMENTAL SCHEMA DISCOVERY APPROACH

We present in this paper an incremental, distributed, density-based clustering algorithm to extract a schema from large and evolving RDF datasets. It allows to keep the schema coherent with the dataset when new entities are added. In order to efficiently manage incrementally growing big datasets, the clustering is restricted to new entities and their neighborhood within the old entities. Clustering new entities and updating the clusters within their neighborhoods ensures that the result is the same as the one provided by the execution of DBSCAN on the global data [3]. Our approach is composed of three main steps described hereafter, and parallelized as illustrated in figure 1.

2.1 Data Distribution

Computing the neighborhood of the new entities may require a very high number of comparisons. In our incremental algorithm, we propose to distribute these new entities over the different processes according to the distribution principle introduced in [1], where the dataset is split into subsets called *chunks* according to the properties of the entities. The intuition behind our distribution method is to group entities sharing some properties into chunks to ensure that all the pairs of similar entities will be detected.

This distribution principle may duplicate the entities in several chunks. To optimize the distribution, we do not consider all the properties of the entities and reduce the cost of the comparison process by skipping useless comparisons.

To this aim, we adapt the notion of *prefix-filter* [2]. The intuition behind this notion is that, in order to be similar, two sets have to share a sufficient number of elements. Thus, We define a *dissimilarity threshold* for an entity e as $dt_\epsilon(e) = |\overline{e}| - \lceil \epsilon \times |\overline{e}| \rceil \quad 1$, considering ϵ as the similarity threshold and $\overline{e}$ as the set of properties of the entity e. This threshold represents the number of properties to consider while distributing the entities.

The clusters that might be updated after the insertion of entities are those within the neighborhood of the new entities. Therefore, the old entities in the neighborhood of a newly inserted entity have to be identified. In our approach, we distribute old entities that share common properties with the new ones over the generated chunks according to our distribution principle.

2.2 Neighborhood Computation

As the chunks contain entities which are likely to be similar, the neighborhood of a new entity is identified by computing the similarity between this new entity and all the other ones in the same

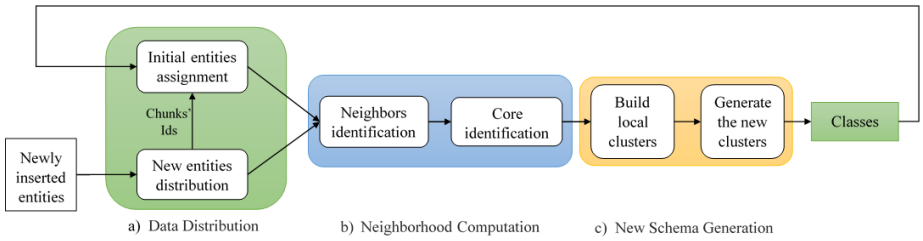


Figure 1: Overview of the Incremental Schema Discovery Approach

chunk. We evaluate the similarity between two entities e_i and e_j using the *Jaccard index*[4]. The comparisons are performed within each chunk in parallel, thus speeding up the clustering process. Since the neighborhood of entities can be distributed over chunks, the neighbors discovered in each chunk are consolidated, and the list of neighbors for each entity in the whole dataset is built. We identify the *core entities* which have a number of neighbors greater or equal to the density parameter *minPts*, and from which the clusters will be initiated. In addition, the insertion of new entities changes the neighborhood of old ones. As a consequence to such change, the clusters existing prior to the insertion of the new entities have to be updated.

2.3 Generating the New Schema

In order to update the schema, we first modify the clusters locally in the chunks based on the neighborhood of the new entities. After adding the set of entities, existing clusters could be updated and new clusters could be created. Based on the core entities, the following change operations can be performed:

- If the neighborhood of a new entity core e' contains an old core entity which belongs to an old cluster C, then e' is assigned to C and C is also expanded with entities that are density-reachable from e'.
- If a core entity e has no old core entity in its neighborhood, then a new cluster is created and the entities that are density-reachable from e are added to this cluster.
- If the neighborhood of a core entity e contains two of more old core entities, which belong to distinct clusters, then these clusters are merged and the resulting cluster is expanded with the entities that are density-reachable from e.
- If an old core entity has a new entity which is not a core within its neighborhood, then the corresponding new entity is absorbed by the cluster containing this old core entity.

These rules are executed in parallel in the different chunks providing local clusters. The final clusters are then built by merging the local clusters which share a common core entity within the newly inserted entities. Finally, the new clusters are added to the old ones to build the new schema.

3 EXPERIMENTS

In our experiments, we evaluate the efficiency of our incremental clustering algorithm compared to the scalable DBSCAN proposed in [1], and derive the speed-up factor of the incremental approach.

Figure 2 illustrates the ability of our incremental algorithm to cluster massive datasets, such as DBpedia, from which we have extracted more than 1.2 million entities. At each insertion, we have added a set of 100k entities to the initial dataset. Then we have compared the execution time of the incremental algorithm to the scalable DBSCAN when executed on the entire dataset.

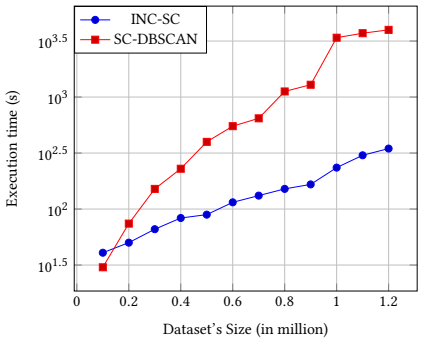


Figure 2: Incremental vs. Sequential Scalable Algorithm

This evaluation shows that the incremental algorithm overcomes the scalable algorithm in terms of performances. In addition, we can observe that the bigger the dataset, the larger the gap between the execution time of both algorithms, and the higher the gain achieved by the incremental approach.

4 CONCLUSION

In this work, we have addressed the problem of incremental evolution of the schema of large RDF datasets as new entities are inserted. We have proposed a novel incremental density-based clustering algorithm which builds groups of similar entities by updating the existing clusters or creating new ones according to the neighborhood of the newly inserted entities. The resulting clusters represent the classes of the schema.

As future works, we could explore the ways of enriching the set of classes with the semantic links between them, and providing semantic annotations for each one. Our algorithms could also be extended in order to exploit partially available schema-related declarations to guide the discovery process, which could improve the quality of the resulting schema.

REFERENCES

[1] Bouhamoum, R., Kedad, Z., and Lopes, S. Scalable schema discovery for RDF data. *Trans. Large Scale Data Knowl. Centered Syst. 46* (2020), 91–120.
[2] Chaudhuri, S., Ganti, V., and Kaushik, R. A primitive operator for similarity joins in data cleaning. In *ICDE* (Atlanta, GA, USA, 2006).
[3] Ester, M., Kriegel, H., Sander, J., Wimmer, M., and Xu, X. Incremental clustering for mining in a data warehousing environment. In *VLDB'98* (1998), pp. 323–333.
[4] Jaccard, P. The distribution of flora in the alpine zone. *New Phytologist 11*, 2 (1912), 37–50.

Une approche déclarative des comparabilités : application aux dépendences fonctionnelles.*

Lhouari Nourine
lhouari.nourine@uca.fr
LIMOS, Université Clermont Auvergne
Aubière, France

Jean-Marc Petit
jean-marc.petit@insa-lyon.fr
LIRIS, Université de Lyon
Lyon, France

Simon Vilmin†
simon.vilmin@ext.uca.fr
LIMOS, Université Clermont Auvergne
Aubière, France

1 PROBLÉMATIQUE

Comment décider que deux valeurs x et y sont égales ? Cette question, simple en apparence, admet en réalité de nombreuses réponses dépendantes de la manière dont l'égalité est définie. En outre, elle se pose implicitement en base de données quand il s'agit de traiter l'inconsistance [3], les données probabilistes [10], les réponses aux requêtes [1, 2, 6], l'intégration de données [7] ou encore le design des bases de données [8]. La question de l'égalité est également cruciale pour la qualité des données.

En pratique, seule l'expertise du domaine permet de donner une définition de l'égalité qui soit adaptée aux données et au contexte. Malheureusement, il n'existe pas à notre connaissance de cadre formel permettant aux experts du domaine de déclarer, à un haut niveau d'abstraction, différentes sémantiques pour l'égalité (prenant en compte les erreurs de mesure, les valeurs nulles, etc.). De ce fait, la gestion de l'égalité est laissée aux programmeurs qui doivent prendre en compte ces différentes possibilités dans leur code.

Dans cet article (voir [9]), nous cherchons à palier ce problème. Nous utilisons les treillis pour introduire un cadre permettant de définir différentes versions de l'égalité à un haut niveau d'abstraction. Ensuite, nous étudions les dépendances fonctionnelles à la lumière de ce cadre déclaratif.

2 CONTRIBUTIONS

D'abord nous introduisons et illustrons le cadre déclaratif général, puis nous résumons les résultats de son application aux dépendances fonctionnelles.

Le contexte associé à un schéma. Considérons un schéma de relation R. Nous attribuons à chaque attribut A de R un *contexte (d'attribut)* comportant une *fonction de comparabilité* f_A et un *treillis de vérité* $\mathcal{L}_A$. La fonction f_A mesure la ressemblance entre les valeurs du domaine de A : elle assigne à chaque paire de valeurs un niveau de similarité représenté par un élément de $\mathcal{L}_A$. En particulier, deux valeurs égales au sens strict (mathématique) du terme ont le plus haut niveau de similarité possible, correspondant à l'élément maximal de $\mathcal{L}_A$. L'apposition des contextes d'attributs donne naissance au *contexte (de schéma)* de R, ainsi qu'à sa fonction de comparabilité f_R. En comparant deux tuples sur R avec f_R, on obtient alors le *tuple abstrait* qui leur est associé. En pratique, ce contexte devrait être donné par les experts du domaine.

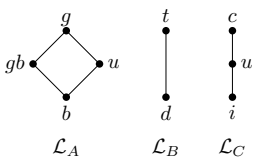
Ensuite, nous introduisons la notion d'*interprétation* du contexte associé à A. Une interprétation est une application h_A qui attribue à chaque élément de $\mathcal{L}_A$ la valeur 0 ou 1. Cette fonction h_A est une sémantique pour l'égalité sur A : deux valeurs u, v du domaine de A sont considérées égales (pour h_A) si $h_A(f_A(x, y)) = 1$. Elles sont différentes sinon. Pour représenter au mieux la notion d'égalité, la fonction h_A doit être croissante, assigner 1 à l'élément maximal de $\mathcal{L}_A$ (le plus haut niveau de similarité possible) et 0 à l'élément minimal de $\mathcal{L}_A$ (le plus faible niveau de similarité). Une *interprétation* g du contexte associé à R résulte de la combinaison d'interprétations pour chacun des attributs de R. Deux tuples sont alors considérés égaux (pour g) si leurs valeurs attributs à attributs sont considérées égales. Ainsi, une interprétation g est une sémantique possible pour l'égalité sur R.

Dans l'article, nous nous intéressons à des interprétations particulières que nous appelons *réalités* et *réalités fortes*. Une réalité est une interprétation qui garantit pour chaque attribut que si deux éléments u et v du treillis de vérité associé sont interprétés à 1, alors $u \wedge v$ est aussi interprété à 1 ($\wedge$-homomorphisme). Intuitivement, une réalité préserve l'opération de ET logique : la conjonction de 1 et 1 est à 1. Une réalité est *forte* si elle respecte aussi l'opération OU : la disjonction de 0 et 0 reste à 0 (homomorphisme).

Example 1. Imaginons que des experts aient fourni des données médicales sur certains patients : leur niveau de triglycéride (en mmol/L), leur sexe (H/F) et leur tour de hanche (en cm). Un extrait de ces données (la relation r) est présenté dans la Figure 1. Par simplicité, on note A, B et C ces trois attributs (resp.) et $R = \{A, B, C\}$.

r	A	B	C
t_1	1.4	F	73
t_2	1.5	F	null
t_3	3.2	M	72
t_4	3.5	F	76
t_5	40	F	100

(a) Une relation r



(b) Treillis de vérité

Figure 1: Une relation r avec les 3 treillis de vérité associés

*. Recherche financée par le gouvernement français, IDEXISITE initiative 16-IDEX-0001 (CAP 20-25)
†. Auteur financé par le CNRS, France, projet ProFan.

De surcroît, les experts ont fourni les fonctions de comparabilités suivantes, pour chacun de ces paramètres :

$$f_A(x,y) = \begin{cases} g & \text{si } x = y \text{ ou } x, y \in [0,2[\\ gb & \text{si } x, y \in [2,5[, x \neq y \\ u & \text{si } (x,y) \text{ ou } (y,x) \in [0,2[\times[2,5[\\ b & \text{sinon.} \end{cases}$$

$$f_B(x,y) = \begin{cases} t & \text{si } x = y \\ d & \text{si } x \neq y \end{cases}$$

$$f_C(x,y) = \begin{cases} c & \text{si } x = y \neq \text{null ou } x, y \in [70,80] \\ u & \text{si } x = \text{null ou } y = \text{null} \\ i & \text{sinon.} \end{cases}$$

Les treillis de vérités associés à ces fonctions sont les treillis $\mathcal{L}_A$, $\mathcal{L}_B$ et $\mathcal{L}_C$ donnés dans la Figure 1. L'ensemble $\{\{f_A, \mathcal{L}_A\}, \{f_B, \mathcal{L}_B\}, \{f_C, \mathcal{L}_C\}\}$ représente donc le contexte du schéma R. Dans la Figure 2 nous illustrons le fonctionnement de notre cadre formel avec les tuples t_2 et t_4. D'abord nous comparons t_2 et t_4 avec f_R et obtenons le tuple abstrait $\langle u, t, u\rangle$. En effet, nous avons $f_A(1.5, 3.5) = gb$, $f_B(F, F) = t$ et $f_C(\text{null}, 76) = u$. Ensuite, nous considérons deux interprétations possibles : g_1 et g_2. Dans la figure, les éléments en blancs (au dessus des pointillés) sont interprétés à 1. Les noirs (en dessous des pointillés) sont interprétés à 0. Il s'en suit que par g_1, les tuples t_2 et t_4 diffèrent sur A et C, tandis ce qu'ils sont égaux pour g_2. Observons que g_1 et g_2 sont des réalités.

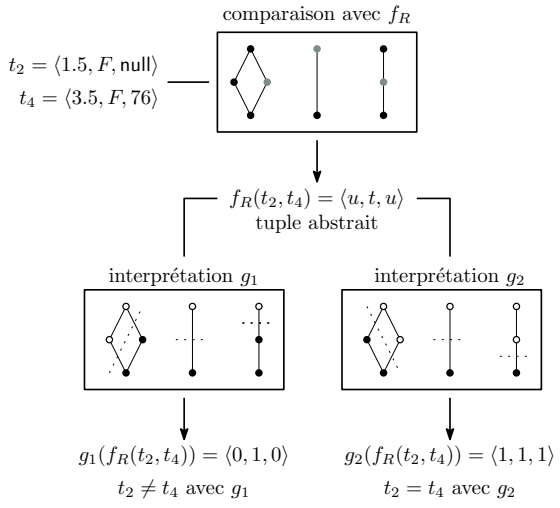


Figure 2: L'égalité de deux tuples avec un contexte de schéma

Application aux dépendances fonctionnelles. À l'aide d'un contexte, il est possible de calculer tous les tuples abstaits d'une relation r. Ceux-ci définissent le *treillis abstrait* $\mathcal{L}_r$ associé à r, qui peut être représenté par un ensemble d'implications de treillis [4], nommées *dépendances fonctionnelles abstraites*. Plus précisément, une dépendance fonctionnelle abstraite est une expression $u \longrightarrow v$ où u et v sont des tuples abstraits issus du produit des treillis de vérité du contexte. Le treillis abstrait associé à la relation r de l'Exemple 1 est illustré dans la Figure 3.

Une interprétation g du contexte devient ainsi une interprétation du treillis abstrait de r. Il apparaît cependant que le résultat $g(\mathcal{L}_r)$ de l'interprétation de $\mathcal{L}_r$ par g peut préserver la sémantique classique des dépendances fonctionnelles (*i.e.* être un système de fermeture sur R [5]) ou non. La question suivante s'en suit : quelles sont les interprétations qui garantissent que l'interprétation de n'importe quel treillis abstrait préserve la sémantique des dépendances fonctionnelles ? Nous montrons que ces interprétations sont précisément les réalités.

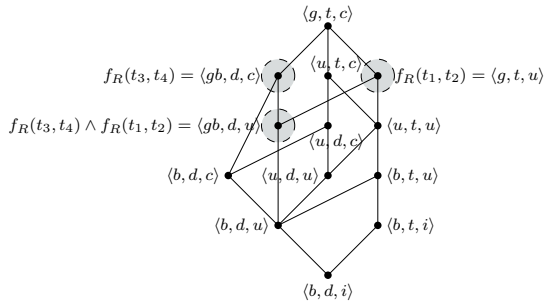


Figure 3: Le treillis abstrait $\mathcal{L}_r$ associé à r

Ensuite, nous étudions plus en détails le lien qui unit treillis abstrait, réalités et dépendances fonctionnelles (dont abstraites). En particulier, nous montrons que si une dépendance fonctionnelle $X \longrightarrow A$ est vraie dans l'interprétation de $\mathcal{L}_r$ par une réalité g, elle est en fait l'interprétation d'une dépendance fonctionnelle abstraite $x \longrightarrow a$ valide dans $\mathcal{L}_r$. Nous prouvons également qu'il existe toujours une réalité qui interprète une dépendance fonctionnelle abstraite $x \longrightarrow a$ valide dans $\mathcal{L}_r$ en une dépendance fonctionnelle $X \longrightarrow A$ valide dans $g(\mathcal{L}_r)$.

Enfin, en s'inspirant des réponses possibles et certaines [1, 2, 6], nous étudions la *plausibilité* d'une dépendance fonctionnelle : existe-t-il une réalité (forte) dans laquelle la dépendance fonctionnelle $X \longrightarrow A$ est vérifiée ? S'il est possible de répondre en temps polynomial pour les réalités en règle générale, le cas particulier des réalités fortes devient NP-complet.

RÉFÉRENCES

[1] Abiteboul, S., Hull, R., and Vianu, V. *Foundations of databases*, vol. 8. Addison-Wesley Reading, 1995.

[2] Amendola, G., and Libkin, L. Explainable certain answers. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (2018), pp. 1683–1690.

[3] Bertossi, L. Database repairing and consistent query answering. *Synthesis Lectures on Data Management 3*, 5 (2011), 1–121.

[4] Day, A. The lattice theory of functional dependencies and normal decompositions. *IJAC 2*, 4 (1992), 409–432.

[5] Demetrovics, J., Libkin, L., and Muchnik, I. B. Functional dependencies in relational databases : A lattice point of view. *Discrete Applied Mathematics 40*, 2 (1992), 155–185.

[6] Greco, S., Pijcke, F., and Wijsen, J. Certain query answering in partially consistent databases. *Proceedings of the VLDB Endowment 7*, 5 (2014), 353–364.

[7] Lenzerini, M. Data integration : A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (2002), pp. 233–246.

[8] Link, S., and Prade, H. Relational database schema design for uncertain data. *Information Systems 84* (2019), 88–110.

[9] Nourine, L., Petit, J. M., and Vilmin, S. Towards declarative comparabilities : application to functional dependencies. *arXiv preprint arXiv :1909.12656* (2019).

[10] Suciu, D., Olteanu, D., Ré, C., and Koch, C. Probabilistic databases. *Synthesis lectures on data management 3*, 2 (2011), 1–180.

goldMEDAL: A Data Lake Generic Metadata Model

Étienne Scholly
Université de Lyon, Lyon 2,
UR ERIC & BIAL-X
Lyon, France
etienne.scholly@bial-x.com

Pegdwendé N. Sawadogo
Université de Lyon, Lyon 2,
UR ERIC
Lyon, France
pegdwende.sawadogo@univ-lyon2.fr

Pengfei Liu
Université de Lyon, Lyon 2,
UR ERIC
Lyon, France
pengfei.liu@eric.univ-lyon2.fr

Javier A. Espinosa-Oviedo
Université de Lyon, Lyon 2,
UR ERIC-LAFMIA lab
Lyon, France
javier.espinosa@imag.fr

Cécile Favre
Université de Lyon, Lyon 2,
UR ERIC
Lyon, France
cecile.favre@univ-lyon2.fr

Sabine Loudcher
Université de Lyon, Lyon 2,
UR ERIC
Lyon, France
sabine.loudcher@univ-lyon2.fr

Jérôme Darmont
Université de Lyon, Lyon 2,
UR ERIC
Lyon, France
jerome.darmont@univ-lyon2.fr

Camille Noûs
Université de Lyon, Lyon 2,
Laboratoire Cogitamus
Lyon, France
camille.nous@cogitamus.fr

ABSTRACT

We summarize here a paper published at the DOLAP 2021 international workshop, which was collocated with EDBT and ICDT. We introduce goldMEDAL, a generic medatata model for data lakes [5].

1 INTRODUCTION

The rise of big data has revolutionized data exploitation practices and led to the emergence of new concepts. Among them, data lakes are large heterogeneous data repositories that can be analyzed by various methods [1].

An efficient data lake requires a metadata system that addresses the many problems arising when dealing with big data. The study of data lake metadata models is currently an active research topic and many proposals have been made [2–4].

However, existing metadata models (including the most recent cited above) are either tailored for a specific use case or insufficiently generic to manage different types of data lakes. To address this issue, we propose goldMEDAL, a generalization of MEDAL, the *MEtadata model for DAta Lakes* [4]. This new metadata model is specified through a classical three-level modeling process, i.e., conceptual, logical and physical.

2 GOLDMEDAL CONCEPTUAL MODEL

goldMEDAL's conceptual conceptual model features four main concepts: data entity, grouping, link and process (Figure 1).

- **Data entities** are the basic units of our metadata model. They are flexible in terms of data granularity. For example, a data entity can represent a spreadsheet file, a textual or semi-structured document, an image, a database table, a tuple or an entire database. The introduction of any new element in the data lake leads to the creation of a new data entity.
- A **grouping** is a set of groups, with a **group** bringing together data entities based on common properties. For example, the raw and preprocessed data zones common in data lake architectures are the groups of a zone grouping. Another example is a grouping of textual documents according to the language of writing.
- **Links** are used to associate either data entities with each other or groups of data entities with each other. They can be oriented or not. They allow the expression of, e.g., simple similarity links between data entities or hierarchies between groups. For example, a temporal hierarchy month → quarter would have the months of January, February and March linked to the first quarter of a given year.
- A **process** refers to any transformation applied to a set of data entities that produces a new set of data entities.

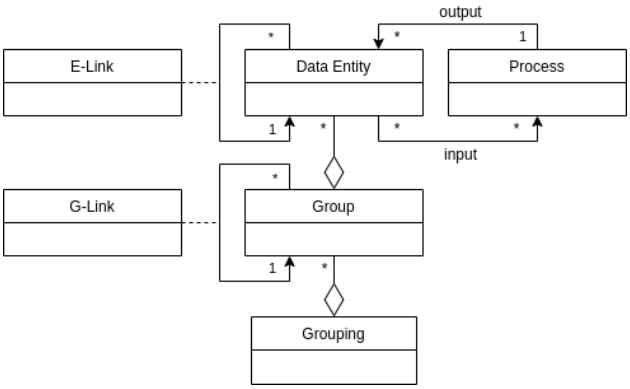


Figure 1: goldMEDAL concepts (UML class diagram)

3 GOLDMEDAL LOGICAL MODEL

At the logical level, goldMEDAL concepts are represented by a graph. Data entities translate into nodes, links translate into edges and groups and processes translate into hyperedges.

For example, in Figure 2, four data entities (say, textual documents) are represented by nodes n_1, n_2, n_3 and n_4. The set of hyperedges $H_1 = \{\theta_{11}, \theta_{12}\}$ represents a *zone* grouping, where θ_{11} and θ_{12} are hyperedges representing the groups *Raw data zone* and *Processed data zone*, respectively. Similarly, $H_2 = \{\theta_{21}, \theta_{22}\}$ represents a *language* grouping, where θ_{21} and θ_{22} represent the groups *French* and *English*, respectively.

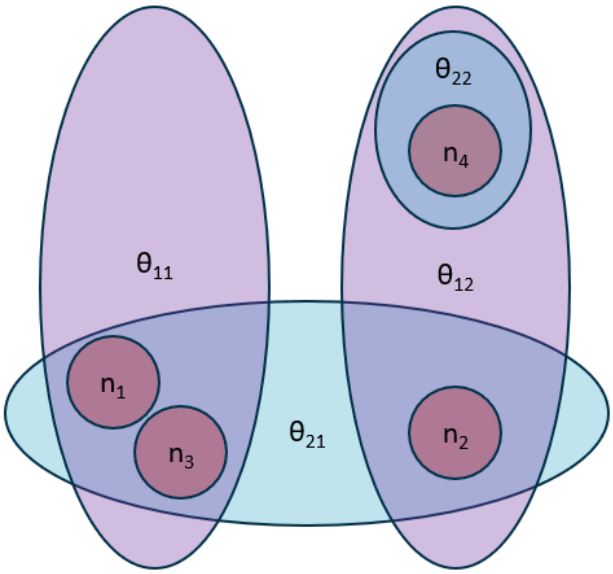


Figure 2: Sample grouping graph logical model

Figure 3 represents the logical model of a process that merges two data entities (say, two relational tables) represented by nodes n_7 and n_8 into a new node n_9. Process $\Pi_1 = \{\Upsilon_1, \Omega_1\}$ is an oriented hyperedge, with $\Upsilon_1 = \{n_7, n_8\}$ and $\Omega_1 = \{n_9\}$ being the sets of input and output nodes of Π_1, respectively.

4 GOLDMEDAL PHYSICAL MODELS

At the physical level, we implemented goldMEDAL into three use-cases/data lakes dedicated to social housing, management sciences and archaeology, respectively. Metadata are managed with dedicated tools (the Neo4J[1] graph database management system or the Apache Atlas[2] data governance and metadata management) or a combination of such tools, i.e., Neo4J, the MongoDB[3] document-oriented database management system (for handling non-atomic metadata) and Elasticsearch[4] search engine (for indexing textual documents).

[1]https://neo4j.com/
[2]https://atlas.apache.org/
[3]https://www.mongodb.com/
[4]https://www.elastic.co/

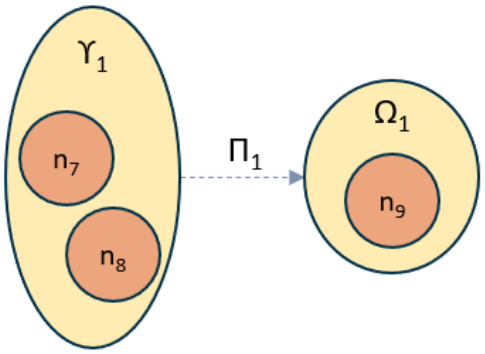


Figure 3: Sample process graph logical model

5 CONCLUSION

Through the three physical models implemented with goldMEDAL, we demonstrated the feasibility and flexibility of our metadata model. Moreover, we demonstrate that goldMEDAL's concepts generalize those of the most recent models from the literature [2–4], which makes of goldMEDAL the most generic metadata model for data lakes as of today.

Another particularity of goldMEDAL is the explicit possibility of data lineage tracing with the concept of process. Thus, goldMEDAL can manage the dynamics of data, while the most recent metadata model, HANDLE [2], does not natively support it.

Future research and open issues include the "industrialization" of data lakes, i.e., providing a software layer, connected to the metadata system, which allows non-data or non-computer scientists to transform and analyze their own data in autonomy.

Furthermore, exploiting a data lake and its metadata system may contribute to open data and open science. A well-designed data lake should indeed readily enforce the four FAIR principles[5].

ACKNOWLEDGEMENTS

Étienne Scholly (PhD), Pegdwendé Nicolas Sawadogo (PhD) and Pengfei Liu's (postdoc) are funded by the BIAL-X company, the AURA Region and the IMU LabEx, respectively.

REFERENCES

[1] Dixon, J. (2010). Pentaho, Hadoop, and Data Lakes. https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/.
[2] Eichler, R., C. Giebler, C. Gröger, H. Schwarz, and B. Mitschang (2020). HANDLE – A Generic Metadata Model for Data Lakes. In *DaWaK 2020*, Volume 12393 of *LNCS*, pp. 73–88.
[3] Ravat, F. and Y. Zhao (2019). Metadata management for data lakes. In *ADBIS 2019*, Volume 1064 of *CCIS*, pp. 37–44.
[4] Sawadogo, P. N., E. Scholly, C. Favre, E. Ferey, S. Loudcher, and J. Darmont (2019). Metadata systems for data lakes: models and features. In *BBIGAP@ADBIS 2019*, Volume 1064 of *CCIS*, pp. 440–451.
[5] Scholly, E., P. N. Sawadogo, P. Liu, J.-A. Espinosa-Oviedo, C. Favre, S. Loudcher, J. Darmont, and C. Noûs (2021). Coining goldMEDAL: A New Contribution to Data Lake Generic Metadata Modeling. In *DOLAP@EDBT/ICDT 2021*, Volume 2840 of *CEUR*, pp. 31–40.

[5]https://www.go-fair.org/fair-principles/

A Hyper-graph Approach for Computing $\mathcal{EL}$ -Ontology Justifications

Hui Yang
yang@lri.fr
LISN, Univ. Paris-Sud, CNRS,
Université Paris-Saclay
Orsay, France

Yue Ma
ma@lri.fr
LISN, Univ. Paris-Sud, CNRS,
Université Paris-Saclay
Orsay, France

Nicole Bidoit
nicole.bidoit@lri.fr
LISN, Univ. Paris-Sud, CNRS,
Université Paris-Saclay
Orsay, France

Justifications are minimal subsets of an ontology that entail a given conclusion. The interest for justification comes from the conciseness of the explanation it provides for a given conclusion. Computing justifications has been widely explored for different tasks, for instance for debugging ontologies [1, 5, 7] and computing ontology modules [4]. For a given conclusion, there may exist more than one justification. Extracting one justification can be easy for tractable ontologies, such as $\mathcal{EL}$ [11]. For instance, we can find a justification by deleting unnecessary axioms one-by-one. However, computing all justifications is complicated and reveals itself to be a challenging problem.

There are mainly two different approaches [11] to compute "all" justifications, the *black-box* approach and the *glass-box* approach. The *black-box* approach [7] relies only on a reasoner and as such can be used for ontologies in any existing Description Logics. For example, a simple (naive) *black-box* approach would check all the subsets of the ontology using an existing reasoner and then filter the subset-minimal ones (i.e., justifications). Many advanced and optimized black-box algorithms have been proposed since 2007 [6]. Meanwhile, the glass-box approaches have achieved better performances over certain specific ontology languages (such as $\mathcal{EL}$ -ontology) by going deep into the reasoning process. Among them, the class of SAT-based methods [1–3, 9, 10] performs the best. The main idea developed by SAT-based methods is to trace, in a first step, a *complete set of inference rules* (*complete set* for short) that contribute to the derivation of a given conclusion, and then, in a second step, to use SAT-based resolution tools to extract all justifications from these inference rules.

In this paper, we improve the state-of-the-art SAT-based approach PULi[9] based on a representation of $\mathcal{EL}$ -ontologies by hyper-graphs. The main idea is to reformulate justifications as special paths called H-paths in the hyper-graph associated with a given ontology. The advantage of this setting is that, most of the time, it reduces the number of the *inference rules* applied to derive a given conclusion, and this accelerates the enumeration of justifications relying on these inference rules. We validate our approach by running real-world ontology experiments. Our hyper-graph based approach outperforms PULi [9], the state of the art algorithm. For instance, for the ontology galen7, for which PULi performed the worst, our method generated ten times fewer inference rules on average and accelerated PULi up to three times. [12].

1 HYPER-GRAPH OF ONTOLOGY

For a given $\mathcal{EL}$ ontology O, the associated hyper-graph $\mathcal{G}_O$= $(\mathcal{V}_O, \mathcal{E}_O)$ is illustrated as Figure 1.

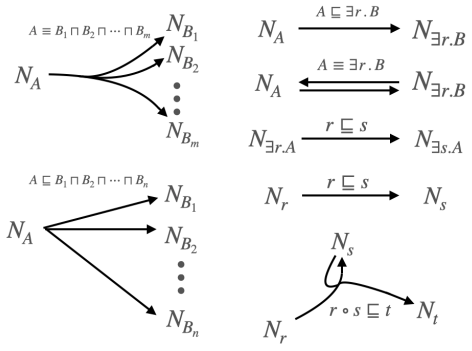


Figure 1: Illustration of hyper-edges.

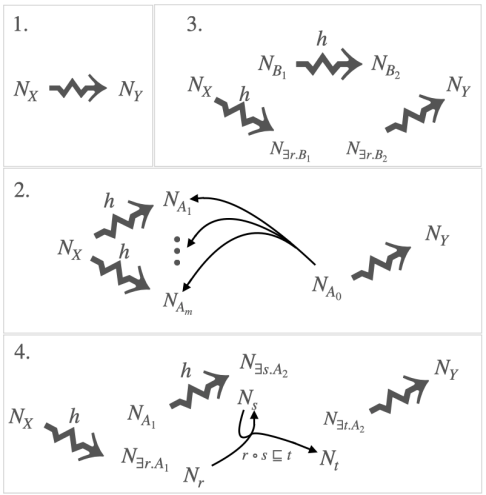


Figure 2: Structure of H-paths ($\rightsquigarrow$ and $\overset{h}{\rightsquigarrow}$ for the existence of simple and H-paths respectively).

As for regular graphs, a simple-path from nodes N_X to N_Y in a hyper-graph is a sequence of edges:

$$e_0=(S_1^0, S_2^0), e_1=(S_1^1, S_2^1), \cdots, e_n=(S_1^n, S_2^n) \quad (1)$$

where $N_X \in S_1^0$, $N_Y \in S_2^n$ and $S_2^{i-1} \cap S_1^i \neq \emptyset, 1 \leq i \leq n$. For the sake of simplicity, next we regard all paths as sets of edges. Then the H-path is defined inductively by using simple-paths and the construction structures as illustrated in Figure 2.

The main result of our study states that the existence of H-path and the ontology entailment match.

Theorem 1. *For any $\mathcal{EL}$ ontology $\mathcal{O}$, $\mathcal{O} \models X \sqsubseteq Y$ iff. there exists a H-path from N_X to N_Y on $\mathcal{G}_\mathcal{O}$.*

Now, we can transpose the problem of finding justification to that of finding minimal H-paths. The algorithm we derive to find minimal H-paths is called minH.

2 EVALUATION

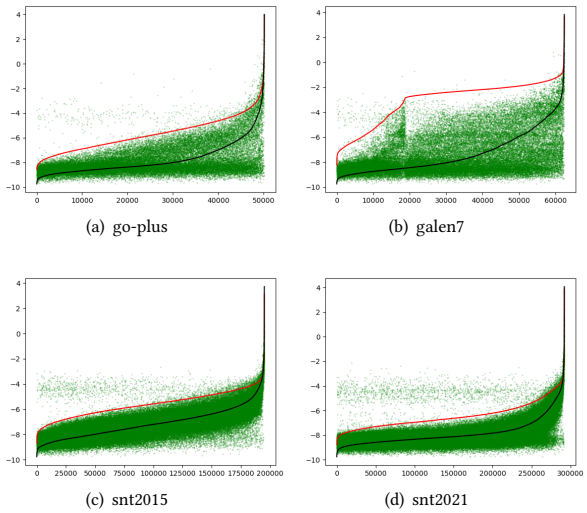
We compare minH with PULi [9] on four different ontologies: go-plus, galen7, snomedct 2015 and snomedct 2021. Both methods compute all justifications based on resolution but with different sets of inference rules generated in different ways. The inference rules of PULi are generated by the ELK reasoner [8] and denoted by *elk*. We call $\mathcal{U}$ the extracted subset of clauses based on our H-rules. To analyze the performance of our setting, we make the following two measures:

- **Size of Inference Rules** We say that a conclusion $A \sqsubseteq B$ is trivial iff all minimal H-paths from N_A to N_B are simple-paths, non-trivial otherwise. The size of inference rules are summarized bellow. It shows that on all four ontologies, $\mathcal{U}$ is much smaller than *elk* on average.

		go-plus	galen7	snt2015	snt2021
elk	average	166.9	3602.0	114.7	67.3
	median	43.0	3648.0	10.0	31.0
	max	7919.0	81501.0	2357	2226
$\mathcal{U}$	average	34.2	74.6	29.4	19.4
	median	4.0	5.0	1.0	3.0
	max	7772	24103	2002	6452
#non-trivial query		50272	62470	195082	304321

Table 1: Summary of size of *elk*, $\mathcal{U}$.

- **Time Cost of Enumerating Justifications** Part of comparisons are illustrated in Figure 3. It shows that minH performs better than PULi, in most of the cases.



(a) go-plus (b) galen7

(c) snt2015 (d) snt2021

Figure 3: The y-axis is the log value of time(s). The red (resp. black) curve presents the increasingly ordered (log value of) time cost of PULi (resp. minH). For a green point (x, y), e^y is the time cost of minH for the query corresponding to the red point (x, y') in the red line.

REFERENCES

[1] M Fareed Arif, Carlos Mencía, Alexey Ignatiev, Norbert Manthey, Rafael Peñaloza, and Joao Marques-Silva. 2016. BEACON: An Efficient SAT-Based Tool for Debugging $\mathcal{EL}^+$ Ontologies. In *International Conference on Theory and Applications of Satisfiability Testing*. Springer, 521–530.

[2] M Fareed Arif, Carlos Mencía, and Joao Marques-Silva. 2015. Efficient axiom pinpointing with EL2MCS. In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*. Springer, 225–233.

[3] M Fareed Arif, Carlos Mencía, and Joao Marques-Silva. 2015. Efficient MUS enumeration of Horn formulae with applications to axiom pinpointing. In *International Conference on Theory and Applications of Satisfiability Testing*. Springer, 324–342.

[4] Jieying Chen, Michel Ludwig, Yue Ma, and Dirk Walther. 2017. Zooming in on Ontologies: Minimal Modules and Best Excerpts. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 10587)*, Claudia d'Amato, Miriam Fernández, Valentina A. M. Tamma, Freddy Lécué, Philippe Cudré-Mauroux, Juan F. Sequeda, Christoph Lange, and Jeff Heflin (Eds.). Springer, 173–189. https://doi.org/10.1007/978-3-319-68288-4_11

[5] Alexey Ignatiev, Joao Marques-Silva, Carlos Mencıa, and Rafael Peñaloza. [n.d.]. Debugging EL Ontologies through Horn MUS Enumeration. ([n. d.]).

[6] Aditya Kalyanpur, Bijan Parsia, Matthew Horridge, and Evren Sirin. 2007. Finding All Justifications of OWL DL Entailments. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007 (Lecture Notes in Computer Science, Vol. 4825)*, Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux (Eds.). Springer, 267–280. https://doi.org/10.1007/978-3-540-76298-0_20

[7] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, and James Hendler. 2005. Debugging unsatisfiable classes in OWL ontologies. *Journal of Web Semantics* 3, 4 (2005), 268–293.

[8] Yevgeny Kazakov, Markus Krötzsch, and Frantisek Simancik. 2012. ELK Reasoner: Architecture and Evaluation.. In *ORE*.

[9] Yevgeny Kazakov and Peter Skočovskỳ. 2018. Enumerating justifications using resolution. In *International Joint Conference on Automated Reasoning*. Springer, 609–626.

[10] Norbert Manthey, Rafael Peñaloza, and Sebastian Rudolph. 2016. Efficient Axiom Pinpointing in EL using SAT Technology.. In *Description Logics*.

[11] Rafael Peñaloza. 2020. Axiom Pinpointing. *arXiv preprint arXiv:2003.08298* (2020).

[12] Rafael Penaloza and Barış Sertkaya. 2017. Understanding the complexity of axiom pinpointing in lightweight description logics. *Artificial Intelligence* 250 (2017), 80–104.

Significance and Coverage in Statistically-Sound Group Testing

Nassim Bouarour
nassim.bouarour@cnrs.fr
CNRS, Univ. Grenoble Alpes
Grenoble, France

Idir Benouaret
idir.benouaret@cnrs.fr
CNRS, Univ. Grenoble Alpes
Grenoble, France

Sihem Amer-Yahia
sihem.amer-yahia@cnrs.fr
CNRS, Univ. Grenoble Alpes
Grenoble, France

ABSTRACT

We tackle the longstanding question of checking hypotheses on the social Web. In particular, we address the challenges that arise in the context of testing an input hypothesis on many data samples, in our case, user groups. This is referred to as Multiple Hypothesis Testing. Ensuring sound discoveries in large datasets poses two challenges: the likelihood of accepting a hypothesis by chance, i.e., returning false discoveries, and the pitfall of not being representative of the input data. We develop GroupTest, a framework for group testing that addresses both challenges. We formulate CoverTest, a generic top-n problem that seeks n user groups satisfying one-sample, two-sample, or multiple-sample tests, and maximizing data coverage. We show the hardness of CoverTest and develop a greedy algorithm with a provable approximation guarantee as well as a more efficient heuristic-based algorithm based on α-investing. Our extensive experiments on multiple real-world datasets demonstrate the necessity to optimize coverage for sound data-driven discoveries, and the efficiency of our heuristic-based algorithm.

KEYWORDS

Exploratory data analysis, hypothesis testing, coverage

1 INTRODUCTION

We develop GroupTest, a unified framework that supports a variety of statistical tests to verify if group behavior supports the null or the alternative hypotheses, and return qualifying groups. We formulate CoverTest, a generic top-n problem that seeks n user groups satisfying one-sample, two-sample, or multiple-sample tests, and maximizing data coverage. CoverTest generalizes the traditional multiple hypothesis problem to accommodate different hypothesis test correction methods and address false discoveries (FWER and FDR). We develop two new algorithms COVER_G and COVER_α to solve CoverTest. COVER_G iterates over the set of candidate groups and chooses the next candidate that maximizes data coverage. Similarly to the traditional FDR procedures, COVER_G controls the false discovery rate at a given significance level α (usually set to 0.05). To do that, COVER_G needs to calculate and rank the p-values of all candidates which prevents it from scaling to a large number of groups. To address that, we propose COVER_α, a heuristic algorithm that builds on α-investing [1], an adaptive sequential method that controls mFDR, the ratio of the expected number of false rejections to the expected number of rejections. The key idea of COVER_α is to invest more significance, referred to as α-wealth, on candidates with the highest coverage. This decision relies on tuning a hyper parameter λ whose value determines the speed at which the α-wealth is consumed, and needs to be explored empirically. COVER_α is efficient and runs in $O(m \cdot n)$, where n is the number of desired results and m is the number of candidates.

2 RELATED WORK

Our work is closely related to multiple hypothesis testing in which a large number of data samples are encountered and need to be considered in testing a given hypothesis. Similarly to existing work on combining data mining, data exploration or machine learning with hypothesis testing, we propose to leverage hypothesis test correction methods to control the risk of false discoveries. Existing work on customer segment discovery [2, 3] combines the computational power of pattern mining with multiple hypothesis testing to find meaningful patterns in the data.

3 PROBLEM

Given a set of users $\mathcal{U}$ and a set of items $\mathcal{I}$, we define *user data* as a database $\mathcal{D}$ of tuples $\langle u, i, a \rangle$ where $a \in \mathbb{R}$ is a value induced by an action such as browsing, tagging, or rating, of user $u \in \mathcal{U}$, on item $i \in \mathcal{I}$. Users have attributes drawn from a set $A_{\mathcal{U}}$ and items have attributes drawn from a set $A_{\mathcal{I}}$. We use $\mathcal{G}$ to denote the set of all groups. Hence, $|\mathcal{G}|$ is the powerset of user and item attribute values.

Problem 1 (CoverTest Problem). *Given a request R, a dataset $D \subseteq \mathcal{D}$ that satisfies user and item conditions, a significance threshold θ on p-values, a number of desired results n, find a set C s.t.*

$$\begin{aligned} &\underset{C \subseteq Candidates}{\texttt{argmax}} \; |cover(\bigcup_{c.groups \in C}, D)| \\ &|C| = n \\ &\textit{subject to} \\ &\forall c \in C \; c.pval \leq \theta \end{aligned} \tag{1}$$

Our problem formulation is generic and aims to accommodate existing significance adjustment procedures by adapting the definition of the significance threshold θ as follows:

- For Bonferroni (BN): $\theta = \frac{\alpha}{m}$
- For Benjamini-Yekutieli (BY): $\theta = \frac{\alpha \times k}{m} \left(\sum_{i=1}^{m} 1/i \right)^{-1}$, where $k = max \left\{ i : p_i \leq \frac{\alpha \times i}{m \times \sum_{i=1}^{m} 1/i} \right\}$, p_i the i^{th} smallest p-value.

where m is the number of groups in *Candidates* and α is the significance level usually set to 0.05

4 ALGORITHMS

To solve CoverTest, we first propose COVER_G (Algorithm 1), a greedy algorithm that scans candidate groups and at each step

Algorithm 1: Greedy coverage algorithm (COVER_G) – illustrated with Benjamini-Yekutieli correction

Input: a Request R, a dataset D, a significance level α, number of desired results n
Output: C

$Candidates \leftarrow GenerateCandidates(R, D)$
$Candidates \leftarrow ComputePvalues(Candidates, \alpha)$
$L = Sortbypval\ (Candidates)$
$C \leftarrow \emptyset$
$m \leftarrow |Candidates|$
$k = \text{argmax}_{0 \le j \le m} L[j] \le \frac{\alpha \times j}{m} \left(\sum_{i=1}^{m} 1/i \right)^{-1}$
while $|C| \le n$ **do**
  $c^* = \text{argmax}_{c \in Candidates}\ cover(C \cup \{c\}, D)$
  $Candidates \leftarrow Candidates \setminus \{c^*\}$
  if $c^*.pval \le L[k]$ **then**
    $C \leftarrow C \cup \{c^*\}$
return C

Algorithm 2: α-investing coverage algorithm (COVER_α)

Input: a request R, a dataset D, a significance level α, number of results n, parameters λ
Output: C

$Candidates \leftarrow GenerateCandidates(R, D)$
$W(0) \leftarrow \alpha$
$\alpha^* = \frac{W(0)}{\lambda\ W(0)}$
$C \leftarrow \emptyset$
$j \leftarrow 1$
while $W(j-1) > 0$ *and* $|C| \le n$ **do**
  $c^* = \text{argmax}_{c \in Candidates}\ cover(C \cup \{c\}, D)$
  $Candidates \leftarrow Candidates \setminus \{c^*\}$
  $\alpha_j = \alpha^* (\frac{|cover(c^*, D)|}{|D|})^{1/2}$
  if $W(j-1) - \frac{\alpha_j}{1-\alpha_j} \ge 0$ **then**
    if $c^*.pval \le \alpha_j$ **then**
      $W(j) \leftarrow W(j-1) \quad \alpha$
      $C \leftarrow C \cup \{c^*\}$
    else
      $W(j) \leftarrow W(j-1) - \frac{\alpha_j}{1-\alpha_j}$
  $j \leftarrow j \quad 1$
return C

selects the candidate with a maximal coverage. The Benjamini-Yekutieli significance adjustment procedure is used to accept or not the selected candidate if its p-value is under the Benjamini-Yekutieli threshold.

We propose a more efficient algorithm COVER_α, that relies on α-investing [4]. The key idea is to re-adjust the quantities of the α-wealth that are invested according to *the coverage* of each selected candidate.

5 EXPERIMENTS

Datasets. We report results on the **MovieLens'1M** dataset. Similar results were observed on other datasets: **Yelp**, **TAFENG**, and **BookCrossing**[1].

Algorithm variants. We compare COVER_G and COVER_α against the traditional correction methods (referred to as TRAD).

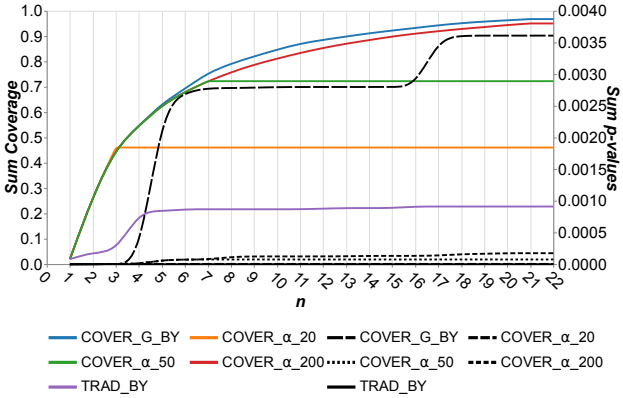


Figure 1: Coverage and p-values as a function of n

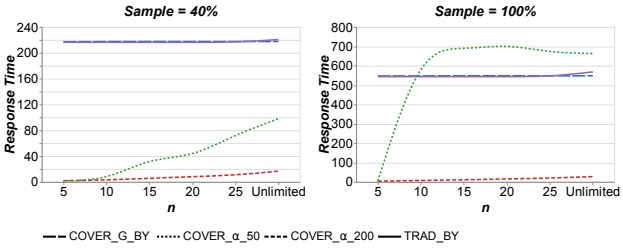


Figure 2: Response time as a function of n

Results. We examine simultaneously the evolution of the cumulative coverage and p-values. Results are depicted in Figure 1. The graph clearly shows that COVER_α_{200} performs closely to COVER_G_BY and reaches full coverage by iteration 22. We also notice that COVER_α_{20} and COVER_α_{50} invest most of the α-wealth in testing insignificant hypotheses. As a result, COVER_α_{50} is no longer capable to add new candidates after its iteration 7.

We study the evolution of response time as a function of number of results n (Figure 2). It shows that COVER_α_{200} clearly outperforms COVER_G and TRAD. Unlike TRAD and COVER_G, COVER_α computes p-values only for candidates with the highest coverage.

Our experiments demonstrate the importance of optimizing coverage in large scale group testing. Our code and complete results are available on our anonymized Github repository.[2]

REFERENCES

[1] D. Foster and R. A. Stine. 2008. Alpha-Investing: A Procedure for Sequential Control of Expected False Discoveries. In *Journal of the Royal Statistical Society: Series B: Statistical Methodology.*
[2] Wilhelmiina Hämäläinen and Geoffrey I. Webb. 2019. A tutorial on statistically sound pattern discovery. *Data Min. Knowl. Discov.* 33, 2 (2019), 325–377.
[3] Leonardo Pellegrina, Matteo Riondato, and Fabio Vandin. 2019. Hypothesis Testing and Statistically-sound Pattern Mining (tutorial). In *KDD 2019, Anchorage, AK, USA, August 4-8, 2019.* 3215–3216.
[4] Zheguang Zhao, Lorenzo De Stefani, Emanuel Zgraggen, Carsten Binnig, Eli Upfal, and Tim Kraska. 2017. Controlling False Discoveries During Interactive Data Exploration. In *SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017.* ACM, 527–540.

[1] http://www2.informatik.uni-freiburg.de/~cziegler/BX/

[2] https://github.com/statistical-group-testing/statistically-soundgrouping

Attaque par inférence d'appartenance sur des séries temporelles agrégées en utilisant la programmation par contraintes

Antonin Voyez
antonin.Voyez@irisa.fr
Univ Rennes, CNRS, IRISA
ENEDIS
France

Tristan Allard
tristan.allard@irisa.fr
Univ Rennes, CNRS, IRISA
France

Gildas Avoine
gildas.avoine@irisa.fr
INSA Rennes, CNRS, IRISA
France

Elisa Fromont
elisa.fromont@irisa.fr
Univ Rennes, CNRS, IRISA
France

Matthieu Simonin
matthieu.simonin@inria.fr
Inria, IRISA
France

Pierre Cauchois
pierre.cauchois@enedis.fr
ENEDIS
France

ABSTRACT

L'agrégation est largement utilisée comme méthode de protection de la vie privée. Les attaques par inférence d'appartenance sur agrégat ont pour but de déterminer si une cible donnée a participé ou non au calcul de l'agrégat attaqué. Dans cet article, nous étudions la vulnérabilé de séries temporelles agrégées - où chaque point est un agrégat horodaté - face à des attaques par inférence d'appartenance. L'attaquant que nous considérons dispose de connaissances auxiliaires sur un sur-ensemble des données agrégées (e.g., issu d'une fuite de données). Nous proposons une nouvelle attaque tirant parti de ce type de connaissances auxiliaires et des multiples points formant la série temporelle agrégat. Notre attaque est modélisée comme un problème d'optimisation linéraire en nombres entiers, permettant à l'attaquant de bénéficier de la puissance des solveurs dédiés (e.g., Gurobi). Cette attaque, testée sur des jeux de données publics, montre la vulnérabilité d'une publication de série temporelle agrégat si le nombre de séries agrégées est trop faible face au nombre de points constituant la série.

Efficient Exploration of Interesting Aggregates in RDF Graphs

Yanlei Diao
yanlei.diao@polytechnique.edu
Ecole Polytechnique, Institut Polytechnique de Paris
Palaiseau, France
Inria
Palaiseau, France

Paweł Guzewicz
pawel.guzewicz@inria.fr
Ecole Polytechnique, Institut Polytechnique de Paris
Palaiseau, France
Inria
Palaiseau, France

Ioana Manolescu
ioana.manolescu@inria.fr
Inria
Palaiseau, France
Ecole Polytechnique, Institut Polytechnique de Paris
Palaiseau, France

Mirjana Mazuran
mirjana.mazuran@inria.fr
Inria
Palaiseau, France
Ecole Polytechnique, Institut Polytechnique de Paris
Palaiseau, France

ABSTRACT

Large (Linked) Open Data are increasingly published as RDF graphs today. However, such data has not yet reached its full potential in terms of sharing and reuse. The main bottleneck here lies in the capacity of human users to explore, discover, and grasp the content and insights of RDF graphs, which are inherently heterogeneous and can be both large and complex. To help democratize the access to such intricate data collections, we need new automatic tools.

We consider the problem of *automatically identifying the k most interesting aggregate queries* that we can evaluate on an RDF graph, given an integer *k* and a user-specified *interestingness function*. Aggregate queries are routinely used to learn insights from relational data warehouses, and some prior research has addressed the problem of automatically recommending interesting aggregate queries. However, the RDF setting presents several differences from the traditional data warehouse setting:

(1) In an RDF graph, we are not *given* but we must *identify* the facts, dimensions, and measures that compose aggregate queries;
(2) Classical relational OLAP algorithms for efficiently evaluating multiple aggregates cannot handle the presence of multi-valued dimensions for a given fact; such dimensions are quite frequently found in RDF data facts and may have zero, one, or more values for dimensions.

To address these challenges, we devise Spade, an *extensible end-to-end framework* that enables the identification and evaluation of interesting aggregates based on MVDCube, our new *RDF-compatible one-pass algorithm for efficiently evaluating a lattice of aggregates*, and a novel *early-stop technique* (with probabilistic guarantees) that prunes uninteresting aggregates and, as a result, reduces the aggregate evaluation cost. Our experiments demonstrate the merit of our approach. Using both real and synthetic graphs, we show the ability of our framework to find interesting aggregates in a large search space, the efficiency of our algorithms (with up to 2.9× speedup over a similar pipeline based on existing algorithms), and scalability as the data size and complexity grow.

This work has originally appeared in SIGMOD 2021 [1].

ACKNOWLEDGMENTS

Yanlei Diao and Paweł Guzewicz are supported by the European Research Council, H2020 research program under GrantNo.: 725561, and by the Agence Nationale de la Recherche under GrantNo.: ANR-16-CE23-0010-01. Mirjana Mazuran is supported by the European Research Council, H2020 research program under GrantNo.: 800192.

REFERENCES

[1] Yanlei Diao, Paweł Guzewicz, Ioana Manolescu, and Mirjana Mazuran. 2021. Efficient Exploration of Interesting Aggregates in RDF Graphs. In *SIGMOD*. Association for Computing Machinery. https://doi.org/10.1145/3448016.3457307

HADAD: A Lightweight Approach for Optimizing Hybrid Complex Analytics Queries

Rana Alotaibi
UC San Diego
ralotaib@eng.ucsd.edu

Bogdan Cautis
University of Paris-Saclay
bogdan.cautis@u-psud.fr

Alin Deutsch
UC San Diego
deutsch@cs.ucsd.edu

Ioana Manolescu
Inria & Institut Polytechnique de Paris
ioana.manolescu@inria.fr

ABSTRACT

Hybrid complex analytics workloads typically include (*i*) data management tasks (joins, selections, etc.), easily expressed using relational algebra (RA)-based languages, and (*ii*) complex analytics tasks (regressions, matrix decompositions, etc.), mostly expressed in linear algebra (LA) expressions. Such workloads are common in many application areas, including scientific computing, web analytics, and business recommendation. Existing solutions for evaluating hybrid analytical tasks – ranging from LA-oriented systems, to relational systems (extended to handle LA operations), to hybrid systems – either optimize data management and complex tasks separately, exploit RA properties only while leaving LA-specific optimization opportunities unexploited, or focus heavily on physical optimization, leaving semantic query optimization opportunities unexplored. Additionally, they are not able to exploit *precomputed (materialized) results* to avoid recomputing (part of) a given mixed (RA and/or LA) computation.

In this paper, we take a major step towards filling this gap by proposing HADAD, an extensible lightweight approach for optimizing hybrid complex analytics queries, based on a common abstraction that facilitates unified reasoning: *a relational model endowed with integrity constraints*. Our solution can be naturally and portably applied on top of pure LA and hybrid RA-LA platforms without modifying their internals. An extensive empirical evaluation shows that HADAD yields significant performance gains on diverse workloads, ranging from LA-centered to hybrid.

The paper appeared in ACM SIGMOD 2021 [1]; it is available online at https://arxiv.org/abs/2103.12317.

ACKNOWLEDGMENTS

This work is supported by a graduate fellowship from KACST.

REFERENCES

[1] R. Alotaibi, B. Cautis, A. Deutsch, and I. Manolescu. HADAD: A lightweight approach for optimizing hybrid complex analytics queries. In G. Li, Z. Li, S. Idreos, and D. Srivastava, editors, *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*, pages 23–35. ACM, 2021.

ASAX : Segmentation adaptative basée sur la quantité d'information pour SAX

Lamia Djebour
INRIA & LIRMM, Univ Montpellier
France
lamia.djebour@inria.fr

Reza Akbarinia
INRIA & LIRMM, Univ Montpellier
France
reza.akbarinia@inria.fr

Florent Masseglia
INRIA & LIRMM, Univ Montpellier
France
florent.masseglia@inria.fr

RÉSUMÉ

Les approches existantes pour le calcul de similitude entre séries temporelles sont au cœur de nombreuses tâches d'analyse de données. Étant donné les volumes de données considérés, ou simplement le besoin de les traiter rapidement, ces approches s'appuient souvent sur des représentations alternatives, plus courtes, qui résument les séries d'origine avec une perte d'information acceptable. Les comparaisons de séries temporelles qui se basent sur ces représentations sont alors approximatives, ce qui fait de la précision un enjeu majeur. Nous présentons et évaluons expérimentalement ASAX, une nouvelle approche pour la segmentation de séries temporelles avant qu'elles soient transformées en représentations symboliques. ASAX réduit de manière significative la perte d'information due aux fractionnements dans les différentes étapes du calcul de la représentation. Nous fournissons des garanties théoriques sur la borne inférieure des mesures de similitude entre séries temporelles, et nos expériences illustrent l'intérêt de notre méthode sur l'état de l'art, en particulier avec un gain de précision significatif.

Cardinality Queries over DL-Lite Ontologies (Extended abstract)

Meghyn Bienvenu
CNRS, University of Bordeaux, Bordeaux INP, LaBRI
Talence, France
meghyn.bienvenu@u-bordeaux.fr

Quentin Manière
CNRS, University of Bordeaux, Bordeaux INP, LaBRI
Talence, France
quentin.maniere@u-bordeaux.fr

Michaël Thomazo
Inria, DI ENS, ENS, CNRS, University PSL
Paris, France
michael.thomazo@inria.fr

ABSTRACT

We summarize our recent work [5] on classifying the complexity of answering cardinality queries over DL-Lite ontologies.

KEYWORDS

Ontology-mediated query answering, Counting queries

A major topic in ontology-mediated query answering (OMQA) research has been to understand the complexity of OMQA and identify tractable settings [6, 11, 12]. Nowadays, for the most commonly considered query language, namely, conjunctive queries (CQs), we have an almost complete picture of the complexity landscape for ontologies formulated in a wide range of different description logics (DLs) [2] and rule-based languages [3, 7]. In particular, it has been shown that CQ answering is tractable in data complexity for ontologies expressed in the most commonly considered dialects of the DL-Lite family [1, 9], which are often employed in OMQA. A crucial property of such DL-Lite dialects and other Horn DLs is that they admit a *canonical model*, which is a single (possibly infinite) model that, by virtue of being homomorphically embeddable into every model, is guaranteed to give the correct answers to all CQs.

While CQs are a natural and well-studied class of queries, there are many other relevant forms of database queries that could be potentially be employed in OMQA. In the present work, our focus will be on counting queries, which together with other forms of aggregate queries, are widely used for data analysis, yet still not well understood in the context of OMQA. A natural way to equip CQs with counting is to count the number of distinct query matches for each answer. As the count value may differ between models, [10] advocated a form of certain answer semantics that considers lower and upper bounds on the count value across different models. Their work provided the first investigation of the complexity of answering counting CQs in the presence of ontologies, revealing such queries to be much more challenging to handle than plain CQs: coNP-complete in data complexity for the well-known DL-Lite$_{core}$ and DL-Lite$^{\mathcal{H}}_{core}$ dialects. A recent work by [4] refined and generalized the complexity results from [10] to a wider class of counting queries and identified a restricted scenario with very low (TC^0-complete) data complexity: rooted CQs coupled with DL-Lite$_{core}$ ontologies. A similar tractability result for connected rooted CQs was proven independently by [8], who also initiated a study of the impact of other restrictions on query shape and developed the first query rewriting procedure for counting CQs. Notably, both the aforementioned TC^0 result and the rewriting procedure crucially relied upon showing that the canonical model gives the right answers under the considered restrictions.

While recent studies have improved our understanding of the complexity of counting CQs, there nevertheless remain many unanswered questions. In this work, we focus on Boolean atomic counting queries of the form $\exists z.\mathrm{A}(z)$ and $\exists z_1, z_2.\mathrm{R}(z_1, z_2)$, which we term *cardinality queries* as they correspond to the natural task of determining (bounds on) the cardinality of a given concept or role name. The data complexity of answering such basic counting queries remains completely open for DL-Lite$_{core}$ ontologies, whilst for DL-Lite$^{\mathcal{H}}_{core}$, the problem is known to be P-hard and in coNP [8]. The main results of our investigation are displayed in Table 1. We show that when ontologies are expressed in DL-Lite$_{core}$, cardinality query answering is tractable in data complexity and enjoys the lowest possible complexity (TC^0-complete). For cardinality queries based upon a concept atom, TC^0 membership holds even for the fragment of DL-Lite$^{\mathcal{H}}_{core}$ obtained by disallowing negative role inclusions. By contrast, for role cardinality queries, we show that coNP-hard situations arise in DL-Lite$^{\mathcal{H}}_{pos}$, which allows only positive concept and role inclusions. In fact, we obtain a complete data complexity classification for DL-Lite$^{\mathcal{H}}_{pos}$, showing that every ontology-mediated query is either TC^0-complete, coNP-complete, or is in P and logspace-equivalent to the complement of Perfect Matching (whose precise complexity is a longstanding open problem). The preceding classification does not extend to DL-Lite$^{\mathcal{H}}_{core}$: we identify new sources of coNP-hardness and further exhibit L-complete cases. We find it intriguing that such complex behaviour arises in what appears at first glance to be a simple OMQA setting. Moreover, in all of the tractable cases we identify, the canonical model may not yield the minimum cardinality, and query answering involves solving non-trivial optimization problems. This led us to devise an entirely new approach based upon exploring a space of strategies to find the optimal way of merging witnesses for existential axioms.

	Concept	Role
DL-Lite$_{core}$	TC^0-c	TC^0-c
DL-Lite$^{\mathcal{H}}_{pos}$	TC^0-c†	TC^0-c | co-PM-c | coNP-c
DL-Lite$^{\mathcal{H}}_{core}$	TC^0-c | L-c | coNP-c | ?	TC^0-c | L-c | co-PM-c | coNP-c | ?

Table 1: Data complexity of cardinality queries based upon concept / role atoms for various DL-Lite dialects. †: holds for all DL-Lite$^{\mathcal{H}}_{core}$ ontologies without negative role inclusions.

ACKNOWLEDGMENTS

Partially supported by ANR project CQFD (ANR-18-CE23-0003)

REFERENCES

[1] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. 2009. The DL-Lite Family and Relations. *Journal of Artificial Intelligence Research (JAIR)* 36 (2009), 1–69.

[2] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. 2017. *An Introduction to Description Logic.* Cambridge University Press.

[3] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. 2011. On rules with existential variables: Walking the decidability line. *Journal of Artificial Intelligence (JAR)* 175, 9-10 (2011), 1620–1654.

[4] Meghyn Bienvenu, Quentin Manière, and Michaël Thomazo. 2020. Answering Counting Queries over DL-Lite Ontologies. In *Proc. of the 29th International Joint Conference on Artificial Intelligence (IJCAI).* 1608–1614.

[5] Meghyn Bienvenu, Quentin Manière, and Michaël Thomazo. 2021. Cardinality Queries over DL-Lite Ontologies. In *Proc. of the 30th International Joint Conference on Artificial Intelligence (IJCAI).* 1801–1807.

[6] Meghyn Bienvenu and Magdalena Ortiz. 2015. Ontology-Mediated Query Answering with Data-Tractable Description Logics. In *Tutorial Lectures of the 11th Reasoning Web International Summer School.* 218–307.

[7] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. 2012. A general Datalog-based framework for tractable query answering over ontologies. *Journal of Web Semantics (JWS)* 14 (2012), 57–83.

[8] Diego Calvanese, Julien Corman, Davide Lanti, and Simon Razniewski. 2020. Counting Query Answers over a DL-Lite Knowledge Base. In *Proc. of the 29th International Joint Conference on Artificial Intelligence (IJCAI).* 1658–1666.

[9] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. 2007. Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family. *Journal of Automated Reasoning (JAR)* 39, 3 (2007), 385–429.

[10] Egor V. Kostylev and Juan L. Reutter. 2015. Complexity of answering counting aggregate queries over DL-Lite. *Journal of Web Semantics (JWS)* 33 (2015), 94–111.

[11] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. 2008. Linking Data to Ontologies. *Journal of Data Semantics* 10 (2008), 133–173.

[12] Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyaschev. 2018. Ontology-Based Data Access: A Survey. In *Proc. of the 27th International Joint Conference on Artificial Intelligence (IJCAI).* 5511–5519.

Cost and Quality Assurance in Crowdsourcing Workflows

Loïc Hélouët, Zoltan Miklos, Rituraj Singh
loic.helouet@inria.fr,zoltan.miklos,rituraj.singh@irisa.fr

Despite recent advances in artificial intelligence and machine learning, many tasks still require human contributions. With the growing availability of Internet, it is now possible to hire workers on crowdsourcing marketplaces. Many crowdsourcing platforms have emerged in the last decade: Amazon Mechanical Turk, Figure Eight[2], Wirk[3], etc. A platform allows employers to post tasks, that are then realized by workers hired from the crowd in exchange for some incentives [3, 19]. Common tasks include image annotation, surveys, classification, recommendation, sentiment analysis, etc. [7]. The existing platforms support simple, repetitive and independent *micro-tasks* which require a few minutes to an hour to complete.

However, many real-world problems are not simple micro-tasks, but rather complex orchestrations of dependent tasks, that process input data and collect human expertize. Existing platforms provide interfaces to post micro-tasks to a crowd, but cannot handle complex tasks. The next stage of crowdsourcing is to build systems to specify and execute complex tasks over existing crowd platforms. A natural solution is to use workflows, i.e., orchestrations of phases that exchange data to achieve a final objective. Figure 1 is an example of complex workflow depicting the image annotation process on SPIPOLL [5], a platform to survey populations of pollinating insects. Contributors take pictures of insects that are then classified by crowdworkers. Pictures are grouped in a dataset D_{in}, input to node p_0 . D_{in} is filtered to eliminate bad pictures (fuzzy, blurred,...) in phase p_0. The remaining pictures are sent to workers who try to classify them. If classification is too difficult, the image is sent to an expert. Initial classification is represented by phase p_1 in the workflow, and expert classification by p_2. Pictures that were discarded, classified easily or studied by experts are then assembled in a result dataset D_{out} in phase p_f, to do statistics on insect populations.

Workflows alone are not sufficient to crowdsource complex tasks. Many data-centric applications come with budget and quality constraints: As human workers are prone to errors, one has to hire several workers to aggregate a final answer with sufficient confidence. An unlimited budget allows hiring large pools of workers to assemble reliable answers for each micro-task, but in general, a client for a complex task has a limited budget. This forces to replicate micro-tasks in an optimal way to achieve the best possible quality, but without exhausting the given budget. The objective is hence to obtain a *reliable* result, forged through a *complex orchestration*, at a *reasonable cost*.

Several works consider data centric models, deployment on crowdsourcing platforms, and aggregation techniques to improve data quality (see [11] for a more complete bibliography). First, coordination of tasks has been considered in languages such as BPMN

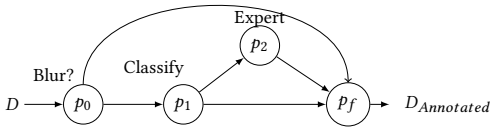


Figure 1: A workflow from SPIPOLL

[18], BPEL [17], or workflow nets [23], a variant of Petri nets dedicated to business processes. They allow parallel or sequential execution of tasks, fork and join operations to create or merge a finite number of parallel threads. Some works propose empirical solutions for complex data acquisition, mainly at the level of micro-tasks [7, 15]. Crowdforge uses Map-Reduce techniques to solve complex tasks [13]. Turkit [16] builds on an imperative language, that allows for repeated calls to services provided by a crowdsourcing platform. Turkomatic [14] implements a Price, Divide and Solve loop, that asks crowd workers to split task into orchestrations of subtasks, and repeats this operation up to the level of micro-tasks.

In this work, we assemble answers returned by workers with so-called *aggregation techniques*. The simplest aggregation is majority voting (MV), a mechanism that considers the most returned answer as ground truth. MV can be improved by giving more weight to competent workers. Other approaches use Expectation Maximization (EM), and consider workers competences to synthesize the most probable correct answer. Competences are expressed in terms of *accuracy* (ratio of correct answers) or in terms of *recall* and *specificity* (that considers correct classification for each possible type of answer). It is usually admitted [26] that recall and specificity give a finer picture of worker's competence than accuracy. Zencrowd [6] uses EM to aggregate answers, and defines competences via accuracy. Workers accuracy and ground truth are hidden variables that must be discovered in order to minimize the deviations between workers answers and aggregated conclusion. D&S [4] uses EM to synthesize answers that minimize error rates from a set of patient records. It considers recall and specificity, but not the difficulty of tasks. [12] proposes an algorithm to assign tasks to workers, synthesize answers, and reduce the cost of crowdsourcing. It assumes that all tasks have the same difficulty, and that workers reliability is a static probability to return a correct value (i.e., the ground truth) that applies to all types of tasks. EM is used by [20] to discover recall and specificity of workers, discover the best experts, and estimate the ground truth. Most of the works cited above consider expertise of workers but do not address tasks difficulty. Approaches such as GLAD [25] or [2] also estimate tasks difficulty to improve quality of answers aggregation on a single batch of Boolean tagging tasks.

A few papers on data aggregation focus on costs optimization. CrowdBudget [22] is an approach that divides a budget B among K existing tasks to replicate them and then aggregate answers with MV. Crowdinc [21] is an EM-based aggregation technique that considers task difficulty, recall and specificity of workers to realize a single batch of micro tasks with a good trade-off between costs and

data quality. It computes accuracy of an aggregation, and launches new tasks dynamically.

Some works consider deployment of tasks, i.e., synthesis of strategies to hire workers and parallelize realization of batches of tasks. The objective is to improve costs and latency, i.e., the time needed to treat a complete batch with an optimal deployment. CLAMSHELL [10] focuses on latency improvement. It affects workers to batches of tagging tasks and detects staggers. To speed up tasks completion, some batches are replicated. Pools are assembled and maintained by rewarding workers for waiting. This approach improves latency, but increases costs. [8] proposes to compute the best static deployment policies in order to achieve an optimal utility (i.e., a weighted sum of overall cost and accuracy) using sequencing or parallelization of tasks. This approach uses a costly exhaustive search which limits the size of deployment problems that can be considered. [24] is a recommendation technique for deployments, that allows parallelization of tasks, sequential composition, and use of machines to solve open tasks such as translation or text writing. This approach builds on optimization techniques to find deployments that reduces latency and improves quality of data.

In this work we propose a solution for the efficient realization of complex tasks. We use a workflow model to orchestrate complex tasks, replicate/distribute them and aggregate the returned results before passing the forged dataset to the next tasks. We extend the complex workflow model of [1], and use the aggregation technique of Crowdinc [21] to forge reliable answers. Our workflow model orchestrates tasks and work distribution according to a dynamic policy that considers confidence in aggregated data and the cost to increase this confidence. A workflow can be seen as an orchestration of phases, where the goal of each phase is to tag records from its input dataset. The output of a phase is used as input for the next ones in the workflow. A complex task terminates when the last of its phases has completed its tagging. For simplicity, we consider simple Boolean tagging tasks that associate a tag in $\{0, 1\}$ to every record in a dataset. Each tagging task on each record is performed by several workers to reduce errors, and the answers are assembled using an aggregation technique. We assume that workers are uniformly paid. For each record, one of the possible answers (the *ground truth*) is correct, and an aggregated answer is considered as reliable if its probability to be the ground truth (computed with aggregation) is high. Hiring more workers to tag records increases the reliability of the aggregated answer. The overall challenge is hence to realize a workflow within a given budget B_0, while guaranteeing that the final dataset forged during the last phase of the workflow has a high probability to be the ground truth.

Design choices influence realization and quality of workflows realization. First, the chosen aggregation technique influences the quality of the final results. Furthermore, the mechanisms used to hire workers impacts costs and accuracy of answers. The simplest way to replicate micro-tasks is *static execution*, i.e., affect an identical fixed number of workers to each micro-task in the orchestration without exceeding budget B_0. On the other hand, one can allocate workers to tasks *dynamically*. One can wait in each phase to achieve a sufficient reliability of answers for all records of the input before forwarding data. This is called a *synchronous* execution of a workflow. Last, one can eagerly forward records with reliable tags to the next phases without waiting for the total completion of a phase. This is called an *asynchronous* execution.

We then study execution strategies for complex workflows in different contexts. We consider several types of workflows, different aggregation mechanisms (namely Majority Voting (MV) and Expectation Maximization (EM) [9]), several distributions of data, difficulty of tasks and workers expertize. We evaluate the cost and accuracy of workflows execution in these contexts under static, synchronous and asynchronous assignment of workers to tasks. Unsurprisingly, dynamic distribution of work saves costs in all cases. A more surprising result is that synchronous realization of complex tasks is in general more efficient than asynchronous realization.

REFERENCES

[1] P. Bourhis, L. Hélouët, Z. Miklos, and R. Singh. Data centric workflows for crowdsourcing. In *Proc. of Petri Nets 2020*, pages 46–61, 2020.
[2] P. Dai, C. H. Lin, and D. S. Weld. Pomdp-based control of workflows for crowdsourcing. *Artificial Intelligence*, 202:52–85, 2013.
[3] F. Daniel, P. Kucherbaev, C. Cappiello, B. Benatallah, and M. Allahbakhsh. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Comput. Surv.*, 51(1):7, 2018.
[4] A.Ph. Dawid and A.M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *J. of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979.
[5] N. Deguines, R. Julliard, M. De Flores, and C. Fontaine. The whereabouts of flower visitors: contrasting land-use preferences revealed by a country-wide survey based on citizen science. *PloS one*, 7(9):e45822, 2012.
[6] G. Demartini, D.E. Difallah, and Ph. Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proc. of WWW 2012*, pages 469–478. ACM, 2012.
[7] H. Garcia-Molina, M. Joglekar, A. Marcus, A. Parameswaran, and V. Verroios. Challenges in data crowdsourcing. *Trans. on Knowledge and Data Engineering*, 28(4):901–911, 2016.
[8] T. Goto, S.and Ishida and D. Lin. Understanding crowdsourcing workflow: Modeling and optimizing iterative and parallel processes. In *Proc. of HCOMP 2016*, pages 52–58. AAAI Press, 2016.
[9] M.R. Gupta and Y. Chen. Theory and use of the em algorithm. *Foundations and Trends in Signal Processing*, 4(3):223–296, 2011.
[10] D. Haas, J. Wang, E. Wu, and M.J. Franklin. Clamshell: Speeding up crowds for low-latency data labeling. *Proc. VLDB Endow.*, 9(4):372–383, 2015.
[11] L. Hélouët, Z. Miklos, and R. Singh. Cost and Quality Assurance in Crowdsourcing Workflows. Extended Version, October 2020.
[12] D.R. Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems. In *Proc. of NIPS'11*, pages 1953–1961, 2011.
[13] A. Kittur, B. Smus, S. Khamkar, and R.E. Kraut. Crowdforge: Crowdsourcing complex work. In *Proc. of UIST'11*, pages 43–52. ACM, 2011.
[14] A. Kulkarni, M. Can, and B. Hartmann. Collaboratively crowdsourcing workflows with turkomatic. In *Proc. of CSCW'12*, pages 1003–1012. ACM, 2012.
[15] G. Li, J. Wang, Y. Zheng, and M.J. Franklin. Crowdsourced data management: A survey. *Trans. on Knowledge and Data Engineering*, 28(9):2296–2319, 2016.
[16] G. Little, L.B. Chilton, M. Goldman, and R.C. Miller. Turkit: tools for iterative tasks on Mechanical Turk. In *Proc. of HCOMP'09*, pages 29–30. ACM, 2009.
[17] OASIS. *Web Services Business Process Execution Language*. 2007.
[18] OMG. *Business Process Model and Notation (BPMN)*. OMG, 2011.
[19] A.J. Quinn and B.B Bederson. Human computation: a survey and taxonomy of a growing field. In *Proc. of SIGCHI'11*, pages 1403–1412, 2011.
[20] V. C. Raykar, S. Yu, L.H. Zhao, G.H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *J. of Machine Learning Research*, 11(Apr):1297–1322, 2010.
[21] R. Singh, L. Hélouët, and Z. Miklós. Reducing the cost of aggregation in crowdsourcing. In *Proc. of ICWS'20*, 2020.
[22] L. Tran-Thanh, M. Venanzi, A. Rogers, and N.R. Jennings. Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks. In *Proc. of AAMAS'13*, pages 901–908, 2013.
[23] W. Van Der Aalst, K. van Hee, A. ter Hofstede, N. Sidorova, H Verbeek, M. Voorhoeve, and M. Wynn. Soundness of workflow nets: classification, decidability, and analysis. *Formal Aspects of Computing*, 23(3):333–363, 2011.
[24] D. Wei, S.B. Roy, and S. Amer-Yahia. Recommending deployment strategies for collaborative tasks. In *Proc. of SIGMOD'20*, pages 3–17. ACM, 2020.
[25] J. Whitehill, T. Wu, J. Bergsma, J.R. Movellan, and P.L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Proc. of NIPS'09*, pages 2035–2043, 2009.
[26] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng. Truth inference in crowdsourcing: Is the problem solved? *Proc. of VLDB Endowment*, 10(5):541–552, 2017.

Shared Processing of Multiple Aggregate Continuous Queries against Spanning and Out-of-Order Events

Aurélie Suzanne
aurelie.suzanne@ls2n.fr
Université de Nantes
Nantes, France
Expandium
Saint-Herblain, France

Guillaume Raschia
guillaume.raschia@ls2n.fr
Université de Nantes
Nantes, France

José Martinez
jose.martinez@ls2n.fr
Université de Nantes
Nantes, France

CCS CONCEPTS

• **Information systems** → **Stream management**; **Query planning**; **Query optimization**.

KEYWORDS

data stream management systems, spanning events, out-of-order events, sliding windows, aggregate continuous queries, multiple query optimization, shared processing

1 INTRODUCTION

Data Stream Management Systems are ubiquitous. They process the huge amount of data generated every day, from our personal devices, as cell phones and IoT, to worldwide transactions, as network traffic for the Internet, stock exchanges or even transportation. Stream processing highly focuses on aggregate computation that provides Key Performance Indicators (KPI) to the end user. The KPIs are expressed as Aggregate Continuous Queries (ACQ) defined by temporal windows.

Nowadays, those streaming systems handle events with a lifespan, such as phone calls, as points in time. They also mainly assume streams with no delay. Both spanning events and out-of-order events undoubtedly yield to noisy aggregates.

Ultimately, multi-ACQs requires near real-time processing and is prone to duplicate computation of aggregates in every query due to overlapping and containment of windows. It then gives the opportunity to save execution cost by sharing sub-aggregates through slicing techniques.

In this communication, we develop an engine for Aggregate Continuous Query (ACQ), which is able to (i) incorporate lifespan to provide exact aggregate computation, (ii) properly manage out-of-order events, and (iii) follow a cost-based policy that elaborates at run-time the most efficient query execution plan of multiple ACQs. The query engine is supported by data structures dedicated to spanning and out-order events and a hybrid sharing schema that aggressively saves computation cost among multiple queries. A lot of experiments have been conducted to show the efficiency of the approach in a large variety of settings and stream profiles.

2 SLICING TECHNIQUE

Up to now, shared execution of ACQs followed two distinct paths: *pooling* or *feeding*. Both approaches are based on the *slicing* technique, which subdivides the windows into non-overlapping time ranges. Each slice then incorporates events in the form of a single partial aggregate. At window release those partial aggregates are combined to form a final aggregate. For a single ACQ, size of the slices is driven by the ACQ's parameters: the *range* defining window size and the *step* representing the window sliding frequency.

3 POOL SHARING

The pooling approach focuses on sharing a common slice set among several ACQs. It requires to create fine-grained slices to fit each and every ACQ. At any time, the system maintains a set of slices that cover all the ongoing windows, as shown in Figure 1, with an obvious slice width of 10 on the $\{a, b, c\}$ example. Such an approach allows reducing insertion cost since only one insertion per pool of ACQs is needed, instead of one insertion per ACQ, but it also increases release cost as slices are smaller.

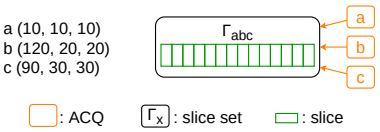


Figure 1: Creation of a slice set for a pool of queries.

The straightforward way of sharing slices is to consider all the ACQs at once, and then build one single slice set. However, if the ACQ steps do not match, it can drastically increase the number of slices required to compute the aggregates. To mitigate that drawback, it is possible to partition the ACQs set into pools of queries. Each pool of ACQs shares a common slice set, while there is no sharing among pools.

Using spanning and out-of-order events in such a pooling configuration is possible, it only requires to insert in several slices for spanning events (as one event may now span over multiple slices), and to scan past slices for out-of-order events. However, without any adaptation, characteristics of such events may also induce errors in results. Indeed, spanning events must be finished in order to release the window. Hence, we add a Time-to-Postpone (TTP) parameter which delays the window release by a fixed amount of time. This TTP is a property of an ACQ, which allows the user to fine tune the delay of each query. For example, one would have a

small TTP for small range ACQs, while a larger TTP is acceptable for longer ranges.

4 FEED SHARING

The feeding approach is based on the idea that in many real-life examples, sliding window queries are not completely unrelated to each other and one ACQ answer might benefit to another ACQ. For example, the aggregates of the sliding window (15 min, 5 min) may be reused by the sliding window (1 hour, 15 min). Thus feed sharing allows feeding a slice set dedicated to an ACQ from the partial results of another slice set, as shown in Figure 2. The main idea is to reduce the number of insertions as the *subscriber* slice sets Γ_b and Γ_c receive a few slices from the *publisher* Γ_a rather than the whole event stream.

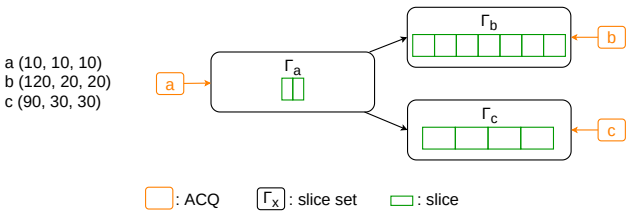


Figure 2: Three slice sets feeding each other. Each slice set is built from its own query step, and it is filled either by the data stream (Γ_a) or by another slice set (Γ_b and Γ_c).

While this idea is quite simple, its application with spanning and out-of-order events needs to be handled with care. Firstly, insertion in a feeding setting with non-delayed point events is always done in the first slice set of the feeding chain. With spanning and out-of-order events, insertion might, however, continue after the first slice set to active subscribing slice sets. Secondly, at window release, slices read will depend on the subscriptions made by a slice set, as an ACQ might read slices only in its slice set, or also in the feeding slice sets.

5 HYBRID SHARING

Pooling and feeding strategies can cohabit such that slice sets are shared among multiple ACQs and subscribe to other slice sets, as a straightforward extension from feeding. An example of such sharing is shown in Figure 3.

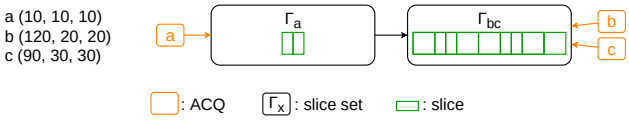


Figure 3: A pooled slice set Γ_{bc} built from queries $\{b, c\}$ and subscribing to another slice set Γ_a.

However, not all combinations are consistent. For two slice sets to be joined they need not to break the subscription chain already in place, and for one slice set to subscribe to another one all its slice bounds must be marked in the feeding slice set.

6 COST ESTIMATION

In order to decide for the best partition, i.e., pooling schema, of the ACQs combined with a relevant feeding schema, one needs to elaborate a cost function that estimates the number of aggregation operations of a specific distribution.

This cost is composed of three parts: insertion, release and shift cost. Insertion cost depends on event size and input rate of the stream. It uses those parameters and the query plan to identify how many slices are used for event insertion per time unit. Release cost uses the range and TTP of the ACQs to count the number of slices read at window release. Finally, the shift cost solely depends on the query plan generated to estimate the number of slices which will be transferred to subscribers at each time unit.

Figure 4 shows a toy example of execution plans given for three queries with different optimization strategies: no-sharing, pool-sharing, feed sharing and the hybrid pool-feed policy. Cost values are given by our cost estimation measure. It also shows promises for the hybrid technique as it exhibits the lowest cost.

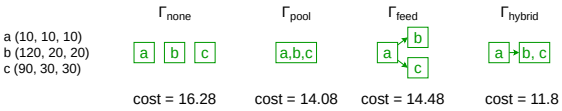


Figure 4: Cost of the query plan following, from left to right, no sharing, pool, feed and hybrid schemes.

As exhaustive search for the best query plan is not possible for medium to large workloads, our system uses a greedy algorithm which starts with all ACQs separated in their own slice set. Then the algorithm tests all the pooling and feeding possibilities between two slice sets and applies the best one. This procedure is done recursively until no option improves the overall cost.

7 EXPERIMENTS

We validate the efficiency of our approach with a set of experiments. In those experiments, we demonstrate the validity of our cost estimation with increasing event size. We also show that under varying settings, our hybrid approach is the most relevant one and provides results better than pooling or feeding approaches.

8 CONCLUSION

In this paper we study multi-ACQ optimization and propose slicing schemes to build efficient query plans in order to leverage shared processing of ACQs. Novelty of that line of work has two dimensions: (i) the assumptions about the data stream: it deals with spanning events and it allows for delays, and (ii) the fully fledged cost measure under those assumptions. Our technique proposes a hybrid schema that supports both pool and feed schemes and adapts to almost any kind of workload.

Évaluez l'Existence d'une Fonction dans votre Jeu de Données avec l'Indicateur g_3

Pierre Faure--Giovagnoli[1,2]
[1]Univ Lyon, INSA Lyon, CNRS, UCBL
LIRIS UMR 5205, Villeurbanne, France
[2]Compagnie Nationale du Rhône
Lyon, France
pierre.faure--giovagnoli@liris.cnrs.fr

Jean-Marc Petit
Vasile-Marian Scuturici
Univ Lyon, INSA Lyon, CNRS, UCBL
LIRIS UMR 5205, Villeurbanne, France
fisrname.lastname@liris.cnrs.fr

RÉSUMÉ

La prise en compte de la connaissance métier dans l'IA est un problème connu, surtout de nos jours où d'énormes quantités de données sont collectées dans l'espoir d'offrir de nouvelles perspectives de valorisation. Considérons le scénario suivant : soit $D(y, x_1, \ldots, x_n)$ un jeu de données, Alice une experte en science des données, Bob un expert du domaine et $y = f(x_1, \ldots, x_n)$ une fonction connue par Bob grâce à sa connaissance du métier. Nous nous intéressons aux questions suivantes, simples mais cruciales pour Alice : comment définir la satisfaction de f dans D ? Quelle est la difficulté de mesurer cette satisfaction ? Il s'avère que ces problèmes sont liés aux dépendances fonctionnelles (DFs) et surtout aux mesures de DFs permettant de quantifier leur satisfaction dans un jeu de données tel que l'indicateur g_3.

Dans cet article, nous étudions le calcul de g_3 avec des DFs strictes et une large classe de DFs non-strictes remplaçant l'égalité stricte par des prédicats plus flexibles. Si le calcul de g_3 avec des DFs strictes peut-être réalisé en temps polynomial, il s'avère être NP-Hard pour les DFs non-strictes. Ainsi, nous proposons différentes solutions exactes et approximées pour le calcul de g_3 pour les deux types de DFs. Tout d'abord, pour les très grands jeux de données, nous proposons des solutions basées sur de l'échantillonnage aléatoire, uniforme et stratifié, pour les DFs strictes. Nous présentons également des algorithmes d'approximation et l'adaptation des progrès récents dans les algorithmes sublinéaires dans le cadre des problèmes NP-Hard pour les DFs non-strictes.

Nous introduisons également FASTG3, une bibliothèque Python open-source qui sera utilisée pour proposer une étude expérimentale approfondie des algorithmes présentés en termes de performances temporelles et de précision d'approximation.

KEYWORDS

contre-exemple, fonction, dépendance fonctionnelle, g3, indicateur, non-stricte, np-hard, données massives, fastg3

1 INTRODUCTION

Lorsqu'elle collabore avec des experts métiers dans le but de confronter leurs connaissances à des données réelles, la spécialiste en science des données a besoin d'un moyen efficace pour exprimer et évaluer leur modèle sur un jeu de données. Souvent, le modèle peut s'exprimer à travers une fonction. Par exemple, la fonction reliant la puissance d'une turbine au débit et à l'élévation du fleuve (et deux constantes ρ et η) comme présenté en Formule 1 nous sert de fil rouge dans la version complète du papier.

$$\text{power} = f_{\eta,\rho}(\text{flow}, \text{elevation}) = \eta \cdot \rho \cdot \text{flow} \cdot \text{elevation} \quad (1)$$

Par conséquent, pour évaluer le bon fonctionnement de leurs turbines et avoir un aperçu des optimisations techniques possibles, il est intéressant d'évaluer la véracité de la fonction 1 par rapport aux données enregistrées sur place. Pour exprimer une telle fonction, les dépendances fonctionnelles (DFs) ont prouvé leur efficacité en offrant un cadre complet pour exprimer des contraintes entre des ensembles d'attributs dans une relation. Par exemple, la fonction 1 peut s'exprimer par la DF suivante :

$$\varphi_{stricte} : \text{flow}, \text{elevation} \rightarrow \text{power}$$

Cependant, cette définition de la satisfaction est connue pour être trop stricte pour de nombreux scénarios de la vie réelle. Ainsi, il est souvent nécessaire de quantifier l'accord partiel d'une DF dans les données plutôt que de se contenter d'évaluer sa satisfaction parfaite pour tous les tuples. La mesure la plus utilisée à cet égard est l'indicateur g_3 [3] avec son opposé $(1 - g_3)$ parfois appelé la confiance. Étant données une relation r et une DF φ, $g_3(\varphi, r)$ correspond à la plus petite proportion de tuples à enlever de r pour que φ soit vérifiée dans r.

De plus, il est intéressant d'étendre l'utilisation de g_3 à d'autres types de DFs. Dans cet article, nous nous concentrons sur les relaxations de la comparaison de valeurs d'attributs où l'égalité stricte imposée par les DFs strictes (aka exacte) est remplacée par des prédicats. Cette importante relaxation permet de capturer une notion plus raffinée de proximité directement liée au problème à résoudre et à la connaissance du domaine disponible. Par exemple, connaissant les incertitudes des capteurs de la turbine, il est possible de proposer une DF non-strictes telle que :

$$\varphi_{nstricte} : [\text{debit}\pm 0.05\cdot\text{debit}], [\text{elevation}\pm 0.05] \rightarrow [\text{puissance}\pm 0.01]$$

Dans ce papier, nous proposons plusieurs algorithmes pour calculer g_3 avec les DFs strictes et non-strictes. Chaque optimisation est détaillée puis comparée à travers une série détaillée d'expérimentations. Ainsi, nous offrons des solutions concrètes permettant de calculer g_3 pour de grands jeux de données, notamment grâce à Fastg3, une nouvelle librairie Python open-source disponible sur GitHub basée sur des implémentations rapides en C++.

2 APERÇU TECHNIQUE

2.1 Dépendances fonctionnelles strictes

Vue générale. Le calcul de g_3 avec DF stricte est similaire à une opération de GROUP-BY. Il s'agit de grouper ensemble tous les tuples ayant les mêmes valeurs sur la partie gauche de la DF et de trouver les tuples ayant une partie droite majoritaire dans chaque groupe. g_3 correspond alors au nombre normalisé de tuples non-majoritaires dans chaque groupe.

Calcul exact. Nous proposons tout d'abord deux algorithmes exacts. L'un est basé sur du hachage avec une complexité linéaire mais un usage mémoire élevé alors que le deuxième utilise un algorithme de tri pour une complexité logarithmique mais un usage mémoire flexible.

Approximation. Pour les grands jeux de données, nous comparons deux algorithmes utilisant les échantillonnages uniforme et stratifié. L'algorithme stratifié est tiré de [1] et nous proposons une amélioration de cet algorithme basée sur une technique d'échantillonnage avec un terme de correction pour population finie permettant des gains importants dans la qualité de l'approximation.

2.2 Dépendances fonctionnelles non-strictes

Vue générale. Notre approche pour le calcul exact de g3 avec une DF non-stricte est composée de deux étapes (généralisation d'un approche proposée dans [5]) : (1) création d'un graphe via l'énumération des paires de tuples ne respectant la DF (aka contre-exemple) et (2) la résolution du minimum vertex cover (MVC) sur le graphe créé. g_3 correspond alors à la taille normalisée de la couverture du MVC. On comprend alors que le calcul de g_3 est coûteux car (1) est quadratique dans le nombre de tuples et (2) requiert la résolution d'un problème NP-Hard de complexité exponentielle dans le nombre de contre-exemples.

Calcul exact. Pour l'opération (1), des optimisations tirées du *record linkage* ou encore des *similarity joins* sont étudiées. Pour la résolution du MVC, Fastg3 propose un algorithme de résolution à l'état de l'art combinant multiples techniques d'optimisation [2].

Approximation. Dans cette étude, nous ne considérons pas d'approximations pour l'opération (1) du calcul. Cependant, une fois le graphe construit, de nombreux algorithmes d'approximation existent pour le MVC permettant d'accélérer l'opération (2). Nous en comparons deux dans les expérimentations : un algorithme avec un très bon ratio d'approximation théorique (2-approché) mais des performances pratiques souvent décevantes et un algorithme avec un mauvais ratio d'approximation théorique mais donnant en pratique des approximations quasi-parfaites.

Approche sublinéaire. Finalement, nous proposons l'alternative des algorithmes sublinéaires pour le MVC permettant d'effectuer les opérations (1) et (2) simultanément et de ne parcourir qu'un sous-ensemble du graphe. Pour ce faire, le graphe est construit à la demande en énumérant les contre-exemples d'un tuple donné uniquement quand c'est nécessaire. Ainsi, en utilisant un système de requête de graphe dissimulant ce système d'énumération à la demande, la plupart des algorithmes sublinéaires peuvent être adaptés pour le calcul du g_3. Nous étudions notamment [6] et [4].

3 EXPÉRIMENTATIONS

Dans le cas des DFs strictes, le calcul de g_3 dépend principalement du nombre de tuples et le temps de calcul reste raisonnable dans l'ordre du million. Néanmoins, les algorithmes d'approximation présentés peuvent être utilisés pour éviter d'itérer sur tous les tuples, permettant ainsi d'accélérer le calcul. Ces algorithmes d'approximation sont particulièrement efficaces pour les petites valeurs de g_3. L'échantillonnage uniforme propose une qualité d'approximation insuffisante mais les algorithmes stratifiés et notamment la version améliorée permettent d'obtenir de très bonnes approximation en un temps raisonnable.

Avec des DFs non-strictes, le calcul de g_3 dépend du nombre de tuples mais aussi du nombre de contre-exemples. Lorsque l'énumération des contre-exemples peut être réalisée en un temps raisonnable, un algorithme du MVC exact peut être utilisé pour calculer exactement g_3 pour un petit nombre de contre-exemples. Sinon, les algorithmes d'approximation du MVC proposent des alternatives efficaces. Lorsque l'énumération devient trop longue, les algorithmes sublinéaires offrent une alternative plus rapide au détriment de la qualité de l'approximation.

4 TRAVAUX FUTURS

Les algorithmes sublinéaires pour le MVC de la littérature adaptent un algorithme 2-approché qui offre, dans les faits, des approximations souvent mauvaises. Nous travaillons sur l'adaptation de l'algorithme GIC en sublinéaire qui permettrait d'obtenir des approximations bien meilleures en pratique.

REMERCIEMENTS

Nous remercions Graham Cormode et Divesh Srivastava pour la qualité de notre échange sur leur article [1]. Merci également à Krzysztof Onak pour avoir aimablement répondu à nos questions sur [4]. Enfin, nous remercions Datavalor de l'INSA Lyon et la Compagnie Nationale du Rhône pour avoir financé une partie de ce travail.

RÉFÉRENCES

[1] Graham Cormode, Lukasz Golab, Korn Flip, Andrew McGregor, Divesh Srivastava, and Xi Zhang. 2009. Estimating the Confidence of Conditional Functional Dependencies. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data* (Providence, Rhode Island, USA) *(SIGMOD '09)*. Association for Computing Machinery, New York, NY, USA, 469–482. https://doi.org/10.1145/1559845.1559895

[2] Demian Hespe, Sebastian Lamm, Christian Schulz, and Darren Strash. 2020. WeGotYouCovered : The Winning Solver from the PACE 2019 Challenge, Vertex Cover Track. In *2020 Proceedings of the SIAM Workshop on Combinatorial Scientific Computing*. SIAM, 1–11.

[3] Jyrki Kivinen and Heikki Mannila. 1995. Approximate inference of functional dependencies from relations. *Theoretical Computer Science* 149, 1 (1995), 129–149.

[4] Krzysztof Onak, Dana Ron, Michal Rosen, and Ronitt Rubinfeld. 2012. A near-optimal sublinear-time algorithm for approximating the minimum vertex cover size. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 1123–1131.

[5] Shaoxu Song. 2010. *Data dependencies in the presence of difference.* Ph. D. Dissertation. Hong Kong University of Science and Technology.

[6] Yuichi Yoshida, Masaki Yamamoto, and Hiro Ito. 2009. An improved constant-time approximation algorithm for maximum˜ matchings. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 225–234.

Lambda+, the renewal of the Lambda Architecture: Category Theory to the rescue

Annabelle Gillet
LIB Univ. Bourgogne Franche Comté
Dijon, France
annabelle.gillet@depinfo.u-bourgogne.fr

Éric Leclercq
LIB Univ. Bourgogne Franche Comté
Dijon, France
eric.leclercq@u-bourgogne.fr

Nadine Cullot
LIB Univ. Bourgogne Franche Comté
Dijon, France
nadine.cullot@u-bourgogne.fr

ABSTRACT

Designing software architectures for Big Data is a complex task that has to take into consideration multiple parameters, such as the expected functionalities, the properties that are untradeable, or the suitable technologies. Patterns are abstractions that guide the design of architectures to reach the requirements. One of the famous patterns is the Lambda Architecture, which proposes real-time computations with correctness and fault-tolerance guarantees. But the Lambda has also been highly criticized, mostly because of its complexity and because the real-time and correctness properties are each effective in a different layer but not in the overall architecture. Furthermore, its use cases are limited, whereas Big Data need an adaptive and flexible environment to fully reveal the value of data. Nevertheless, it proposes some interesting mechanisms. We present a renewal of the Lambda Architecture: the Lambda+ Architecture, supporting both exploratory and real-time analyzes on data. We propose to study the conservation of properties in composition of components in an architecture using the category theory.

KEYWORDS

Architecture pattern, Category theory, Lambda Architecture

1 INTRODUCTION

All information systems have a common point: they need an architectural design before being developed and deployed. The architecture must guarantee some properties and guide the consistency of the overall structure of the information system. In this context, architectural styles and patterns are used to build a system having the expected characteristics for each of its part as well as for its entirety, and to state the requirements of the technologies and programming techniques needed to achieve the goal sought. Thus, global requirements such as scalability, performance, reliability must be clearly identified to select the style of architecture, the different components and the interactions among them [6], and then choose technologies with properties (such as ACID for databases or micro batch capabilities for stream processing) that fit all of the previous choices. The absence of coherence in a definition of an architecture can lead to the dreaded Big Ball of Mud [3], that reduces greatly the maintenance and evolutivity capabilities of the system.

Recent researches in software architecture try to formally define styles and patterns, to anticipate effects of the composition of components, and thus knowing beforehand the result of the evolution of a part of the architecture [1, 5]. When architectures evolve and grow, they can combine several smaller parts of architectures developed separately. When building a large scale, complex and distributed architecture, its parts can embed architecture styles on their own. These different cases can result in compositions of smaller architecture parts with their proper styles and patterns, so formalization should be able to express and control these compositions. Category theory [2] is a promising approach for formalization, due to its ease to represent compositions as it considers morphims and functors as first class citizens, and to its already existing proximity to the engineering software world, particularly with functional programming. Moreover, its graphical representation is a visual help to understand the formalization, and leads to a better comprehension of the system [9].

We propose the Lambda+ Architecture pattern, an update of the Lambda Architecture, and a formalization to study the conservation of properties in compositions of components using the category theory. For more details on this work, see [4].

2 THE LAMBDA ARCHITECTURE PATTERN

The properties of correctness, low latency and fault-tolerance have always been a major concern when designing architectures. In [6], Lampson sketches some suggestions that are still relevant today, and that can be found, among others, in the Lambda Architecture, introduced by Marz in 2011 [7, 8]. The objective of the Lambda is simple: to compute predetermined queries with a very low latency and to ensure the correctness of the processing. To do so, the Lambda is composed of three layers: the batch layer, that takes care of storing raw data in the master dataset and of executing the computations on the batch of data while preserving the correctness property, the speed layer, that performs the same computations as the batch layer but with an incremental processing to support the low latency property but not the correctness one, and the serving layer, that puts the results to disposal.

With these specifications, the advantages of the Lambda Architecture are a strong fault-tolerance for machine and human faults, a guarantee of a correct result with the batch layer and a low latency with the speed layer. However, the Lambda has also been criticized a lot, due to its complexity to maintain and to evolve both the speed and the batch layers, that have to perform the same computations, but with different paradigms. It also lacks in flexibility, as its goal is to answer only predetermined queries. Thus, alternative use cases

such as exploratory analyzes require to modify the pattern. Furthermore, by delegating the correctness property only to the batch layer and not to the speed layer, the low latency and the correctness properties cannot be obtained simultaneously. To clarify this statement, the Lambda has to be replaced in the context of its creation. At this time, streaming systems were only at their early stages, and thus did not have all the capabilities that they have today. It includes the correctness property, that have since been integrated into the stream processing systems. So, the Lambda can be seen as a mean to compensate flaws of an emerging technology, rather than a pattern that fully exploits it.

3 THE LAMBDA+ ARCHITECTURE PATTERN

To improve the Lambda Architecture, the correctness property should hold for all the components. Furthermore, the fault-tolerance should be kept, but as the reprocessing of data in a batch fashion is incompatible with the real-time property, it should be integrated as an alternative running composition of components, activated only in case of a technical failure or to satisfy new needs. Use cases should also gain in flexibility, and the complexity induced by the development of the same process with different paradigms in different layers should be avoided.

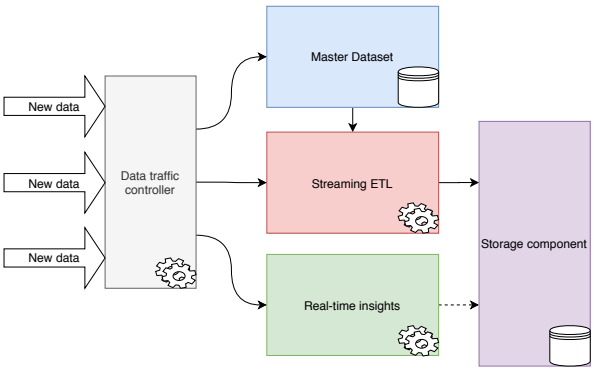


Figure 1: Overview of the Lambda+ Architecture

The Lambda+ Architecture (figure 1) is meant to be a renewal of the Lambda Architecture, by improving the support of the correctness property and by leveraging two main functionalities: 1) storing data in a way that allows flexible and exploratory data analyzes ; and 2) computing in real-time predefined queries on data streams in order to have insights on well-known and identified needs. The duality between exploratory analyzes and predefined queries is of primary importance in a Big Data context, where the combination of volume and variety of data overcomes the capability of finding all the insights hidden in data. The fault-tolerance mechanism of the Lambda is kept, but is only activated when needed.

The Lambda+ is composed of a set of components interacting together asynchronously with messages. This pattern borrows its principles from the Event-Driven Architecture style, which is well-suited for achieving performance, scalability and evolutivity. The trade-offs of this architecture style is a lack of simplicity and the difficulty of testing the whole architecture, due to the dynamic nature of the messaging workflow and the chaining of various processing components.

4 USING THE CATEGORY THEORY TO STUDY CONSERVATION OF PROPERTIES

In the research field of software engineering for architecture design, the need for proper theory and formalization has raised importance in the last decade [1, 5]. Designing, specifying and implementing software architectures are complex tasks, that require careful specifications to link and preserve characteristics through all the steps of creation. The development of theory in this field requests both practical and theoretical skills, in order to propose a model suited to the expectations, that takes into consideration the imperfections of the real-world of engineering.

To fill this need, category theory [2] is a promising approach: it allows to switch from a model to another or to navigate among abstraction levels [9]. By focusing on relations (the morphisms) and compositions, it proposes powerful mechanisms that can be applied to architectures: the behaviour of functors combined with preorders allows the study of the conservation or the discarding of properties in compositions of components.

5 CONCLUSION

We proposed the Lambda+ Architecture pattern, the successor of the Lambda Architecture, that gets rid of its flaws and fits more various use cases by handling both exploratory and real-time analyzes. We used the category theory to study the conservation of properties in compositions of components.

For future work, we plan to develop our formalization to study more various aspects of architectures: 1) to navigate among abstraction levels (i.e., the level of detail of the representation of the architecture) ; 2) to verify if an architecture follows a given style or pattern by using full functors (i.e., surjective functors) ; and 3) to extend the property description, including numerical values (e.g., the execution time to deduce if it can be considered as real-time).

ACKNOWLEDGMENTS

This work is supported by ISITE-BFC (ANR-15-IDEX-0003) coordinated by G. Brachotte, CIMEOS Laboratory (EA 4177), University of Burgundy.

REFERENCES

[1] Manfred Broy. 2011. Can practitioners neglect theory and theoreticians neglect practice? *Computer* 44, 10 (2011), 19–24.
[2] Samuel Eilenberg and Saunders MacLane. 1945. General theory of natural equivalences. *Trans. Amer. Math. Soc.* 58, 2 (1945), 231–294.
[3] Brian Foote and Joseph Yoder. 1997. Big ball of mud. *Pattern languages of program design* 4 (1997), 654–692.
[4] Annabelle Gillet, Éric Leclercq, and Nadine Cullot. 2021. Lambda+, the Renewal of the Lambda Architecture: Category Theory to the Rescue. In *International Conference on Advanced Information Systems Engineering*. Springer, 381–396.
[5] Pontus Johnson, Mathias Ekstedt, and Ivar Jacobson. 2012. Where's the theory for software engineering? *IEEE software* 29, 5 (2012), 96–96.
[6] Butler W Lampson. 1983. Hints for computer system design. In *Proceedings of the ninth ACM symposium on Operating systems principles*. 33–48.
[7] Nathan Marz. 2011. How to beat the CAP theorem. http://nathanmarz.com/blog/how-to-beat-the-cap-theorem.html
[8] Nathan Marz and James Warren. 2015. *Big Data: Principles and best practices of scalable real-time data systems.* Manning.
[9] David I Spivak. 2014. *Category theory for the sciences.* MIT Press.

Threats Modeling And Anomaly Detection In The Behaviour Of A System - A Review Of Some Approaches

Meriem Ghali
meriem.ghali@etu.univ-lyon1.fr
Université Lyon 1
France

Crystalor Sah
crystalor.sah@etu.univ-lyon1.fr
Université Lyon 1
France

Marie Le Guilly
marie.le-guilly@univ-lyon1.fr
LIRIS UMR 5205 CNRS
Université Lyon 1
France

Mohand-Saïd Hacid
mohand-said.hacid@univ-lyon1.fr
LIRIS UMR 5205 CNRS
Université Lyon 1
France

ABSTRACT

With the increase of Big Data, cybersecurity is undergoing massive changes, especially when attacks process differently and use different strategies.
Because of the vast volume of data, it becomes harder and harder to detect anomalies, it is complex to manage with traditional systems, and therefore to devise techniques to automatically identify malicious behaviours, even though it is a crucial task. However, Big Data also enables the development of new anomaly detection approaches, based on data analysis and especially machine learning and data mining. With this perspective, it becomes possible to propose solutions that are more flexible and better suited to the new threats that are constantly evolving. In this paper, our objectives are to first define some concept in cybersecurity, describe a model for identifying computer security threats, a model for assessment and prevention tool with CyberKill Chain (CKC), give a general overview of the current techniques used for anomaly detection in the context of cybersecurity. Then, implement and test some machine learning techniques for this task, in order to compare their performances. Experiments were carried on the CICIDS2017 dataset, using traditional anomaly detection techniques based on clustering algorithms like K-Means and Classification algorithms such as SVM, decision tree, and neural networks algorithms.
To better understand the section presented in our paper, **Algorithm 1** describes the process and the main idea we wanted to convey :

Algorithm 1: Process of the paper

```
if Threat then
    try:
        Detect vulnerabilities
        Delete vulnerabilies
        Secure the system ;
    end
    /* Detect at which level the attack operates using CKC  */
    Classify the threat on a stage of the CyberKill Chain
    if Threat operates at the system level then
        Use anomaly detection algorithms
        if Avaible data then
            Use Supervised algorithms
        else if One labeled Avaible data then
            Use Semi-Supervised algorithms
        else
            Use Unpervised algorithms
        end
    else
else
    Monitor logs to predict the normal behavior and detect
     future threats
end
```

KEYWORDS

Anomaly Detection, Data Mining, Intrusion Detection Systems, Clustering, Classification, Threat Modeling, Attack Modeling

Tractable Orders for Direct Access to Ranked Answers of Conjunctive Queries

Nofar Carmeli
Nofar.Carmeli@ens.fr
ENS, PSL University, France

Nikolaos Tziavelis
tziavelis.n@northeastern.edu
Northeastern University, USA

Wolfgang Gatterbauer
w.gatterbauer@northeastern.edu
Northeastern University, USA

Benny Kimelfeld
bennyk@cs.technion.ac.il
Technion, Israel

Mirek Riedewald
m.riedewald@northeastern.edu
Northeastern University, USA

ABSTRACT

We study the question of when we can provide logarithmic-time direct access to the k-th answer to a Conjunctive Query (CQ) with a specified ordering over the answers, following a preprocessing step that constructs a data structure in time quasilinear in the size of the database. Specifically, we embark on the challenge of identifying the tractable answer orderings that allow for ranked direct access with such complexity guarantees.

We begin with *lexicographic orderings* and give a decidable characterization (under conventional complexity assumptions) of the class of tractable lexicographic orderings for every CQ without self-joins. We then continue to the more general *orderings by the sum of attribute weights* and show for it that ranked direct access is tractable only in trivial cases. Hence, to better understand the computational challenge at hand, we consider the more modest task of providing access to only a single answer (i.e., finding the answer at a given position) — a task that we refer to as *the selection problem*. We indeed achieve a quasilinear-time algorithm for a subset of the class of full CQs without self-joins, by adopting a solution of Frederickson and Johnson to the classic problem of selection over sorted matrices. We further prove that none of the other queries in this class admit such an algorithm.

CCS CONCEPTS

• **Theory of computation → Database theory**; *Complexity classes*; *Database query languages (principles)*; *Database query processing and optimization (theory)*.

KEYWORDS

conjunctive queries, direct access, ranking function, answer orderings, query classification

Compatibility Checking Between Privacy and Utility Policies: A Query-Based Approach

Hira Asghar
Christophe Bobineau
firstname.lastname@univ-grenoble-alpes.fr
Université Grenoble Alpes, CNRS, Grenoble INP, LIG
Grenoble, France

Marie-Christine Rousset
Marie-Christine.Rousset@univ-grenoble-alpes.fr
Université Grenoble Alpes, CNRS, Grenoble INP, IUF, LIG
Grenoble, France

ABSTRACT

Data sharing over the internet through smart devices is susceptible to disclose sensitive information of data producers. To protect the privacy of data producers, we propose a query-based approach where data producers keep their data on decentralized personal data servers and only disclose data to data consumers over secure communication links according to their privacy policies. In our approach, we express the privacy and utility policies as sets of temporal aggregated conjunctive queries, and we study compatibility between privacy and utility policies based on their query expressions.

KEYWORDS

temporal RDF graphs, temporal aggregated conjunctive queries, utility policy, privacy policy

1 INTRODUCTION

Personal data are increasingly disseminated over Internet through mobile devices and smart environments, and are exploited for developing more and more sophisticated services and applications. All these advances come with serious risks for privacy breaches that may reveal private information wanted by users to remain undisclosed. It is therefore of utmost importance to help data producers to keep the control on their data for their privacy protection while preserving the utility of disclosed data for service providers.

In this paper, we approach the problem of utility-aware privacy preservation in the setting of applications where service providers (e.g., power suppliers) perform data analytics on data concerning their customers (e.g., smart home occupants) for optimization or recommendation purposes. In such settings, (sensor) data are gathered, abstracted and transferred through internet protocols from data producers environment (e.g., smart home, smart personal devices) to a centralized data consumer in charge of aggregating data for conducting varied analytics tasks.

Sensitive data leakage can occur at different stages and places due to security vulnerabilities of *(1)* the network, *(2)* the centralized server used by the data consumer for collecting data outsourced by the different data producers, and *(3)* the local servers of each data producer.

Following the vision of [1], we propose, first, to rely on data encryption to secure data exchange through the network and, second, to avoid the privacy risks of data centralization by keeping the data produced by each data owner decentralized in secure personal data servers.

The approach that we promote to face the privacy versus utility dilemma in this setting can be summarized as follows:

(1) Data producers keep the control on the data they accept to transmit to the data consumer according to their own *privacy policy*.
(2) The data consumer makes explicit *his/her utility policy* to explain for which task or service s/he requests data from data producers.
(3) In case of incompatibility of the utility policy with the privacy policy of a data producer, the data producer negotiates with the data consumer to find an acceptable privacy-utility trade-off.

Our contribution is twofold. First, we extend the framework proposed in [3] to *formalize privacy and utility policies as temporal aggregate queries*. Second, we formally define and study the compatibility problem in this query-based framework. More details can be found in [2].

2 QUERY-BASED SPECIFICATION OF POLICIES

We define utility and privacy policies in the form aggregated conjunctive queries that are built upon a common temporal knowledge graph.

Definition 1 (Temporal aggregated conjunctive query)**.** A *TACQ* is defined as
SELECT $\bar{x}$, $agg(y)$
WHERE $\{GP\ .\ FILTER\}$
GROUP BY $\bar{x}$
TIMEWINDOW $(Size, Step)$
where

- GP is a temporal graph pattern,
- $FILTER$ is a boolean combination of atomic comparisons of the form $t > t'$ or $t \geq t'$ where t and t' are variables of $Var(GP)$ or literals (numbers, strings or dates),
- $\bar{x}$ is a tuple of variables called the output (or grouping) variables,
- when the aggregate term $agg(y)$ is present, y (called the aggregate variable) is not in $\bar{x}$ and agg is an aggregate function that produces a single value when applied to a set of values assigned to y.

- *Size* and *Step* are time durations (i.e. differences between timestamps),

Definition 2 (Utility policy)**.** A utility policy is defined by a set of TACQ queries, called utility queries. A utility policy (issued by a service provider) is satisfied by a data producer if s/he accepts to provide the set of answers of all of the utility queries for any RDF graph storing her/his data.

Definition 3 (Privacy policy)**.** A privacy policy is defined by a set of TACQ queries, called privacy queries. A privacy policy, specific to each data producer, is satisfied when all the answers of all the privacy queries remain undisclosed for any temporal RDF graph storing her/his data.

3 COMPATIBILITY CHECKING

Incompatibility between privacy and utility policies should be defined and checked independently of any data graph and should express that no answer of a privacy query can be inferred from answers to some utility queries. This is formalized in Definition 4 in which the *logical signature* of an answer to a query is the logical formula built from the query expression that characterizes all the (unknown) data graphs leading to this answer (for this query).

Definition 4 (Incompatibility between privacy and utility)**.** A privacy policy is incompatible with a utility policy iff the logical signature of an answer to a privacy query can be logically entailed by the union of logical signatures of answers sets of utility queries.

Theorem 3.1 provide a full characterization of incompatibility for privacy and utility queries without aggregate. It is based on building all the data graphs that are representative of the different ways of joining answers of utility queries. Each of these data graph is obtained by $freezing$ the variables in the union of graph patterns in the utility queries, in a way that allows to replace distinct output variables with a same constant.

Theorem 3.1 (Incompatibility without aggregate). *A privacy query without aggregate is incompatible with a set of utility queries without aggregate if and only if there exists a freezing of the union of the graph patterns in the utility queries on which the evaluation of the privacy query returns a non-empty set of answers.*

In the worst case, this may lead to iterating an exponential number of times the evaluation of each privacy query over small RDF graphs, since the number of possible freezings is 2^{OV_u} where OV_u is the number of output variables in the utility queries.

In the following, we give sufficient conditions of (in-)compatibility that are easier to check.

Theorem 3.2 (Sufficient condition of compatibility). *A privacy query is compatible with a set of utility queries if*

- *the graph pattern of the privacy query is disjoint from the union of the graph patterns of the utility queries,*
- *or if the conjunction of the FILTER condition of the privacy query and of all the FILTER conditions in the utility queries is unsatisfiable,*
- *or if no boundary of a time window of any utility query corresponds to a boundary of a time window of the privacy query, except of course for the first ones starting at query evaluation time.*

For the case of privacy and utility queries with aggregates, since the aggregate values are computed based on groups that are specific to each query, checking compatibility can be restricted to checking compatibility between each privacy query and each utility query.

Theorem 3.3 (Sufficient condition of incompatibility). *Let consider a privacy query Q_p and an utility query Q_u with aggregates on a same time window. Q_p is incompatible with Q_u if there exists a (possibly empty) freezing f_p of output variables in GP_p with constants of GP_u, or a (possibly empty) freezing f_u of output variables in GP_u with constants in GP_p such that $f_p(GP_p)$ and $f_u(GP_u)$ are isomorphic.*

When Q_p and Q_u have no FILTER conditions and the same aggregate functions, they are incompatible if and only if the above condition is satisfied.

4 CONCLUSION AND FUTURE WORK

In this paper we have proposed a query-based declarative framework for a formal specification and verification of privacy and utility policies expressed as temporal aggregate conjunctive queries.

We do think that this framework is well suited for guaranteeing data producers to keep the control and protect their data in many real-world situations where sensitive data are collected by mobile personal devices or smart environments.

Based on the implementation of this framework, we plan to design and implement a negotiation mechanism that will be triggered when a utility policy turns out to be incompatible with a privacy policy. New relaxed utility queries will be automatically computed to restore compatibility with the privacy policy of a given data producer. They will be the formal basis of a dialogue between each data producer and the service provider in order to find a trade-off acceptable in terms of utility while guaranteeing privacy preservation for each data producer.

We also plan to extend our framework to take into account ontological knowledge in the possible inference of answers of privacy queries by answers of utility queries. This will bring stronger constraints on compatibility between privacy and utility policies.

5 ACKNOWLEDGMENTS

This work has been partially supported by MIAI@Grenoble Alpes (ANR-19-P3IA-0003), PERSYVAL-Lab (ANR-11-LABX-0025-01) and TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215.

REFERENCES

[1] Tristan Allard, Nicolas Anciaux, Luc Bouganim, Yanli Guo, Lionel Le Folgoc, Benjamin Nguyen, Philippe Pucheral, Indrajit Ray, Indrakshi Ray, and Shaoyi Yin. 2010. Secure personal data servers: a vision paper. *Proceedings of the VLDB Endowment* 3, 1-2 (2010), 25–35.

[2] Hira Asghar, Christophe Bobineau, and Marie-Christine Rousset. 2021. *Compatibility Checking Between Privacy and Utility Policies: A Query-Based Approach.* Research Report. Université Grenoble Alpes ; CNRS ; Grenoble INP ; Laboratoire d'informatique de Grenoble. https://hal.archives-ouvertes.fr/hal-03385977

[3] Remy Delanaux, Angela Bonifati, Marie-Christine Rousset, and Romuald Thion. 2018. Query-Based Linked Data Anonymization. In *The Semantic Web-ISWC 2018* (Monterey, California, United States). Springer, Cham, 530–546. https://doi.org/10.1007/978-3-030-00671-6_31

Processing SPARQL Property Path Queries Online with Web Preemption

Julien Aimonier-Davat
Université de Nantes, LS2N
Nantes, France
julien.aimonier-davat@univ-nantes.fr

Hala Skaf-Molli
Université de Nantes, LS2N
Nantes, France
hala.skaf@univ-nantes.fr

Pascal Molli
Université de Nantes, LS2N
Nantes, France
pascal.molli@univ-nantes.fr

RÉSUMÉ

Les requêtes SPARQL property path représentent un outil indispensable pour chercher des motifs complexes dans un graphe de connaissance. Cependant, évaluer ces requêtes sur des données en ligne, et obtenir des résultats complets, est difficile. Du fait de leur complexité, les requêtes property path sont souvent intérrompues par les politiques d'usage équitable en place sur les SPARQL endpoints. Pour garantir des résultats complets, le client Web décompose les requêtes property path en un ensemble de sous-requêtes, dont la terminaison est garantie par le serveur. La granularité de la décomposition dépend de l'expressivité du serveur. Quand bien même, décomposer une requête property path peut générer un important traffic entre le client et le serveur, si bien que le temps d'exécution d'une requête se retrouve dominé par les coûts réseaux. Dans ce papier, nous étendons le modèle de la préemption Web avec un nouvel opérateur, capable d'évaluer des fermetures transitives partielles (PTC). Cet opérateur repose sur l'utilisation d'un algorithme de recherche en profondeur, dont l'exploration est limitée à une profondeur k, définie à l'avance. Nous montrons ensuite qu'un client Web, s'il dispose d'un serveur SPARQL préemptif sur lequel est implémenté l'opérateur PTC, est capable d'exécuter efficacement n'importe quelle requête property path, avec la garantie d'obtenir des résultats complets. Une étude expérimentale confirme que, par rapport aux approches par décomposition, notre approche réduit drastisquement la quantité de données et le nombre d'appels échangés entre le client et le serveur, et ainsi, le temps d'exécution des réquêtes SPARQL property path.

Reasoning in $\mathcal{EL}$-Description Logic with Refreshing Variables

Théo Ducros
LIMOS
Aubière, France
theo.ducros@limos.fr

Marinette Bouet
LIMOS
Aubière, France
marinette.bouet@uca.fr

Farouk Toumani
LIMOS
Aubière, France
farouk.toumani@limos.fr

ABSTRACT

Description Logics (DLs) are a family of knowledge representation and reasoning formalisms that have been proven useful in many application domains[2]. They provide means for well-structured and formal representation of the conceptual knowledge of an application domain and various inference procedures to reason about the represented knowledge. Description logics enable to describe a universe of discourse in terms of *concept descriptions*, i.e., expressions built from atomic concepts (unary predicates) and atomic roles (binary predicates) using the constructors of the considered description logic. For example, using the atomic concept *Person* and the atomic role *haschild*, the concept of *Parent* can be represented by the concept description

$$Person \sqcap \exists haschild.Person$$

While subsumption reasoning, i.e., the ability to determine subconcept–superconcept relationships, is a traditional reasoning in DL-based system additional inference mechanisms, such as matching [4] and unification [3], that go beyond subsumption, have been proposed in the literature. In this latter non-standard forms of reasoning, concept description with not completely specified form (incomplete information) can be specificed using the so-called concept patterns, i.e., concept descriptions containing variables. As an example, consider the following pattern which defines an *Academic* as a *Person* with a certain relationship with a *University*.

$$Academic \equiv Person \sqcap \exists x.University$$

Here, the variable x takes its values from a set of possible atomic role names. The concept description

$$Academic \equiv Person \sqcap \exists worksIn.University$$

matches this pattern. Indeed, if we replace the variable x by the role *worksIn*, the pattern becomes equivalent to the description. Replacing a variable x with a value is called variable substitution. Given a description C and a pattern P, the matching problem asks then whether there is a variable substitution such that C matches P. The unification extends the matching to the case where C is itself a pattern. Consider now the case of a cyclic description:

$$Academic \equiv Person \sqcap \exists x.University \sqcap \exists y.Academic$$

and the following concept descriptions:

$$A \equiv Person \sqcap \exists worksIn.University \sqcap \exists hasAdviser.B$$

$$B \equiv Person \sqcap \exists presidentOf.University \sqcap \exists hasChild.B$$

Using standard semantics of variable substitution, A do not match the pattern academic. However, if we exploit a different semantics that enables to *refresh* the values of the variable x and y in each description of the pattern *Academic*, in this case it becomes possible to compute a matcher that makes the concept A match the pattern *Academic* (i.e., the first occurrences of x and y are respectively mapped to *worksIn* and *hasAdviser* while their second occurrences are respectively mapped to *presidentof* and *hasChild*).

This paper studies the extension of description logics with variables equipped with refreshing semantics. More specifically, we focus on a new description logic, called $\mathcal{EL}_V$, that extends the description logic $\mathcal{EL}$ with refreshing variables. Our definition of $\mathcal{EL}_V$-patterns deviates from the one used in the literature with respect to the following features:

(1) our definition of concept patterns use role variables while the literature mainly focuses on concept variables,
(2) we support cyclic pattern definition and allow two different types of semantics for variables (i.e., refreshing and not refreshing semantics),

We consider in particular a new reasoning mechanism in this context, called *weak-subsumption*, which extends matching and unification to logics with refreshing variables. Our main technical result is to show that testing weak-subsumption between $\mathcal{EL}_V$-patterns with role variables is EXPTIME-complete. Our approach to test weak-subsumption exploits the link between subsumption and the simulation relation between the so-called description graphs introduced in [1]. The main steps of our approach are as follows:

- We associate with each $\mathcal{EL}_V$-pattern P a description automaton A_P. This automata corresponds to a compact representation of all possible substitutions of P.
- We extend the notion of simulation relation, used in [1] to characterize weak-subsumption between $\mathcal{EL}$-patterns. Our main technical results consists in characterizing weak-subsumption between $\mathcal{EL}_V$-patterns in terms of existential simulation between $\mathcal{EL}_V$-description automata.
- We devise an algorithm to test existential simulation between $\mathcal{EL}_V$-description automata and prove its correctness. We show its optimality by demonstrating that the proposed algorithm has exponential time complexity in the worst case.

REFERENCES

[1] Franz Baader. 2003. Terminological cycles in a description logic with existential restrictions. In *IJCAI*, Vol. 3. 325–330.
[2] Franz Baader, Diego Calvanese, Deborah McGuinness, Peter Patel-Schneider, Daniele Nardi, et al. 2003. *The description logic handbook: Theory, implementation and applications*. Cambridge university press.
[3] Franz Baader and Barbara Morawska. 2009. Unification in the Description Logic $\mathcal{EL}$. In *International Conference on Rewriting Techniques and Applications*. Springer, 350–364.
[4] Franz Baader and Barbara Morawska. 2014. Matching with Respect to General Concept Inclusions in the Description Logic $\mathcal{EL}$. In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*. Springer, 135–146.

Creating Embeddings of Heterogeneous Relational Datasets for Data Integration Tasks

Riccardo Cappuzzo
cappuzzo@eurecom.fr
EURECOM

Paolo Papotti
papotti@eurecom.fr
EURECOM

Saravanan Thirumuruganathan
sthirumuruganathan@hbku.edu.qa
QCRI, HBKU

The problem of data integration concerns the combination of information from heterogeneous relational data sources, which is recognized as an expensive task for humans [12]. While traditional approaches require substantial effort from domain scientists to generate features and labeled data or domain specific rules, there has been increasing interest in achieving accurate data integration with deep learning methods to reduce the human effort. Embeddings have been successfully used for this goal in data integration tasks such as entity resolution [1, 4, 8, 10, 11], schema matching [6, 9], identification of related concepts [5], and data curation in general [12]. Typically, these works fall into two dominant paradigms based on how they obtain word embeddings. The first is to reuse *pre-trained* word embeddings computed on a generic corpus for a given task. The second is to build *local* word embeddings that are specific to the dataset. These methods treat each tuple as a sentence by reusing the same techniques for learning word embeddings employed in natural language processing.

However, both approaches fall short in some circumstances. Enterprise datasets contain custom vocabulary, as in the small datasets in the left-hand side of Figure 1. The pre-trained embeddings do not capture the *local* semantics expressed by these datasets and do not contain embeddings for the word "Rick". Also, the embedding for token "Steve" is closer to tokens "iPad" and "Apple" even though it is not implied in the data. Approaches that treat a tuple as a sentence miss a number of signals such as attribute boundaries, integrity constraints, and so on. Moreover, existing approaches do not consider the generation of embeddings from heterogeneous datasets, with different attributes and alternative value formats. These observations motivate the generation of *local* embeddings for the *relational* datasets at hand.

We advocate for the design of such local embeddings that leverage both the *relational nature* of the data and the downstream task of *data integration.*

Tuples are not sentences. Embedding techniques originally developed for *textual* data ignore the richer set of semantics inherent in *relational* data. Consider a cell value $t[A_i]$ of an attribute A_i in tuple t, e.g., "Mike" (in italic) in the first relation from the top. Conceptually, it has semantic connections with other attributes of tuple t (such as "iPad 4th") and other values from the domain of attribute A_i (such as "Paul", also in italic in the figure).

Embedding generation must span different datasets. Embeddings must be trained using heterogeneous datasets, so that they can meaningfully leverage and surface similarity across data sources. A notion of similarity between different types of entities, such as tuples and attributes, must be developed. Tuple-tuple and attribute-attribute similarity are important features for data integration.

Any solution satisfying these requirements must overcome multiple challenges. First, it is not clear how to encode the semantics of the relational datasets in the embedding learning process. Second, datasets may share limited amount of information, have different schemas, and contain a different number of tuples. Finally, datasets are often incomplete and noisy. The learning process is affected by low information quality resulting in embeddings that do not correctly represent the semantics of the data.

We address these challenges with EmbDI, a framework for building relational, local embeddings for data integration that introduces a number of innovations to effective model heterogeneous tuples. We identify crucial components and propose effective algorithms for instantiating each of them. EmbDI is designed to be modular so that anyone can customize it by plugging in other algorithms and benefit from the continuing improvements from the deep learning and database communities. The two main components in our solution are the following.

1. Graph Construction. We use a compact tripartite graph-based representation of relational datasets that effectively represents syntactic and semantic data relationships. Specifically, we use three types of nodes. *Token* nodes correspond to the unique values found in the dataset. *Record Id* nodes (RIDs) represent a unique token for each tuple. *Column Id* nodes (CIDs) represent a unique token for each column/attribute. These nodes are connected by edges based on the structural relationships in the schema. This graph is a compact representation of the original datasets that highlights overlap and explicitly represent the primitives for data integration tasks, i.e., records and attributes.

2. Embedding Construction. We formulate the problem of obtaining local embeddings for relational data as a graph embeddings generation problem. We use random walks to quantify the similarity between neighboring nodes and to exploit metadata such as tuple and attribute IDs. This method ensures that nodes that share similar neighborhoods will be in close proximity in the final embeddings space. The corpus that is used to train our local embeddings is generated by materializing these random walks.

In this short paper, we give an overview of the solution and report results for the entity resolution task. We refer the reader to the extended version for more details and full experimental results [2].

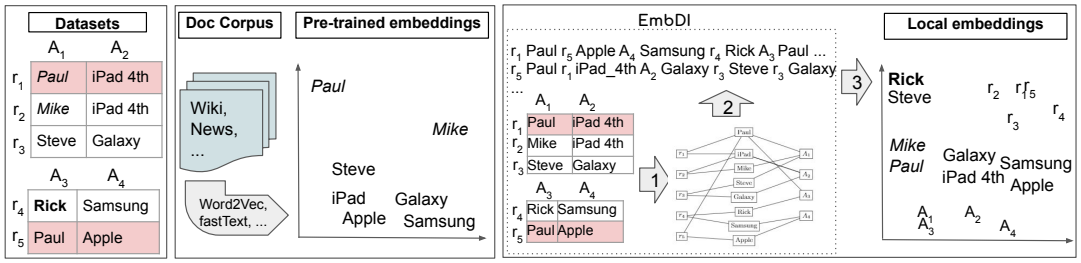


Figure 1: A vector space learned from text (prior methods) and from data (EmbDI).

Overview

Our framework, EmbDI, consists of three major components, as depicted in the right-hand side of Figure 1.

(1) In the *Graph Construction* stage, we transform the relational dataset in a compact tripartite graph that encodes various relationships inherent in it. Tuple and attribute ids are treated as first class citizens.

(2) Given this graph, the next step is *Sentence Construction* through the use of biased random walks. These walks are carefully constructed to avoid common issues such as rare words and imbalance in vocabulary sizes. This produces as output a series of sentences.

(3) In *Embedding Construction*, the corpus of sentences is passed to an algorithm for learning word embeddings. Depending on available external information, we optimize the graph and the workflow to improve the embeddings' quality.

Experiments

We used 8 datasets from the literature and a dataset with a larger schema (IM) that we created from open data (https://www.imdb.com/interfaces/, https://grouplens.org/datasets/movielens/). For the majority of the scenarios, less than 10% of the distinct data values are overlapping across the two datasets.We relied on state of the art methods to combine words in tuples and to obtain embeddings for words that are not in the pre-trained vocabulary [4].

We test four algorithms for the generation of local embeddings. All methods make use of our tripartite graph and exploit record and column IDs in the integration tasks. The first method (Basic) creates embeddings from permutations of row tokens and sentences with samples of attribute tokens. The second method (N2V [7]) is a widely used algorithm for learning node representation on graphs. Given our graph as input, it learns vectors for all nodes. The third method is Harp [3], a state of the art algorithm that learns embeddings for graph nodes by preserving higher-order structural features. The fourth method is EmbDI (https://gitlab.eurecom.fr/cappuzzo/embdi), with walks (sentences) of size 60, 300 dimensions for the embeddings space and the Skip-Gram model in word2vec with a window size of 3. We also use different tokenization strategies to convert cell values in nodes (Simple, Flatten and Overlap whose details are reported in the full paper). As baseline for this *unsupervised* case, we use our matching algorithm with pre-trained embeddings obtained from fastText (fTxt).

Results in Table 1 for unsupervised settings show that EmbDI-O embeddings obtain the best quality results in three scenarios and second to the best in four cases. In every case, local embeddings obtained from our graph outperform pre-trained ones.

	Pretrain	Local				
	fTxt	EmbDI-S	EmbDI-F	EmbDI-O	N2V	Harp
BB	0.59	0.50	0.82	**0.86**	**0.86**	**0.86**
WA	0.58	0.59	0.75	**0.81**	mem	0.78
AG	0.18	0.14	0.57	0.59	0.70	**0.71**
FZ	0.99	0.98	0.99	0.99	**1.00**	**1.00**
IA	0.10	0.09	0.09	0.11	mem	0.14
DA	0.72	0.95	0.94	0.95	0.87	**0.97**
DS	0.80	0.85	0.75	**0.92**	mem	0.81
IM	0.31	0.90	0.64	0.94	mem	**0.95**

Table 1: F-Measure results for Entity Resolution (ER).

Acknowledgement. This work has been partially supported by the ANR JCJC grant *ANR-18-CE23-0019* and by the IMT *Futur & Ruptures* program "AutoClean".

REFERENCES

[1] Öykü Özlem Çakal, Mohammad Mahdavi, and Ziawasch Abedjan. 2019. CLRL: Feature Engineering for Cross-Language Record Linkage. In *EDBT*. 678–681.
[2] Riccardo Cappuzzo, Paolo Papotti, and Saravanan Thirumuruganathan. 2020. Creating Embeddings of Heterogeneous Relational Datasets for Data Integration Tasks. In *SIGMOD*. ACM.
[3] Haochen Chen, Bryan Perozzi, Yifan Hu, and Steven Skiena. 2017. HARP: Hierarchical Representation Learning for Networks. *CoRR* abs/1706.07845 (2017). arXiv:1706.07845 http://arxiv.org/abs/1706.07845
[4] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. 2018. Distributed representations of tuples for entity resolution. *PVLDB* 11, 11 (2018), 1454–1467.
[5] Raul Castro Fernandez and Samuel Madden. 2019. Termite: a system for tunneling through heterogeneous data. *arXiv preprint arXiv:1903.05008* (2019).
[6] Raul Castro Fernandez, Essam Mansour, Abdulhakim A Qahtan, Ahmed Elmagarmid, Ihab Ilyas, Samuel Madden, Mourad Ouzzani, Michael Stonebraker, and Nan Tang. 2018. Seeping semantics: Linking datasets using word embeddings for data discovery. In *ICDE*.
[7] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *SIGKDD*. ACM, 855–864.
[8] Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. 2019. Low-resource Deep Entity Resolution with Transfer and Active Learning. *arXiv preprint arXiv:1906.08042* (2019).
[9] Christos Koutras, Marios Fragkoulis, Asterios Katsifodimos, and Christoph Lofi. 2020. REMA: Graph Embeddings-based Relational Schema Matching. *SEA Data workshop* (2020).
[10] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *SIGMOD*.
[11] Rohit Singh, Venkata Vamsikrishna Meduri, Ahmed K. Elmagarmid, Samuel Madden, Paolo Papotti, Jorge-Arnulfo Quiané-Ruiz, Armando Solar-Lezama, and Nan Tang. 2017. Generating Concise Entity Matching Rules. In *SIGMOD*. ACM, 1635–1638.
[12] Saravanan Thirumuruganathan, Nan Tang, Mourad Ouzzani, and AnHai Doan. 2020. Data curation with Deep Learning. *EDBT* (2020).

La qualité des outils de l'analyse de sentiments : les raisons de l'incohérence

Wissam Mammar Kouadri
Université de Paris
France
wissam.maamar_kouadri@etu.u-paris.fr

Mourad Ouziri
Université de Paris
France
mourad.ouziri@u-paris.fr

Salima Benbernou
Université de Paris
France
salima.benbernou@u-paris.fr

Karima Echihabi
Mohammed VI Polytechnic University
Maroc
karima.echihabi@um6p.ma

Themis Palpanas
LIPADE, Université de Paris
France
themis@mi.parisdescartes.fr

Iheb Ben Amor
IMBA Consulting
France
iheb.benamor@imbaconsulting.com

ABSTRACT

Dans ce papier, nous présentons une étude empirique profonde pour évaluer et étudier le phénomène d'incohérence intra et inter-algorithmes dans les outils d'analyse de sentiment. L'étude couvre plusieurs axes (statistique, structurel et sémantique), pour déterminer les causes et les facteurs qui provoquent les incohérences. Un benchmark de test a été crée et une heuristique pour affiner sa qualité a été proposée. Nos résultats ont montré que les incohérences sont fréquentes dans toutes les catégories d'algorithmes d'analyse de sentiment existants.

KEYWORDS

Analyse de sentiment, résolution des incohérences, qualité de données

1 INTRODUCTION

L'analyse de sentiment est le processus d'extraction de la polarité d'un texte afin de déterminer son orientation sémantique : positive, négative, ou neutre. Grâce à sa présence dans plusieurs domaines de recherche [7, 14], beaucoup travaux [2, 4, 5, 13, 15, 17] se sont penchés sur l'analyse de sentiment et des outils performants pour l'extraction automatique de la polarité du texte. Néanmoins, malgré les avancées de la recherche réalisées dans ce domaine, cette tâche reste difficile à cause de la richesse du langage naturel et la dépendance de la polarité au contexte. Pour illustrer cette complexité, considérons les deux phrases : (a) Donald Trump softens tone on Chinese investments et (b) Trump drops new restrictions on China investment.

Bien que (a) et (b) soient structurées différemment, elles sont sémantiquement équivalentes et expriment la même idée. Plusieurs travaux de recherche [2, 4, 5, 13, 15, 17] ont conclu que les textes sémantiquement équivalents ont la même polarité. En revanche, des travaux récents comme [3, 9, 10, 12] ont montré que les outils d'analyse de sentiment ne respectent pas ce consensus et attribuent des polarités différentes aux paraphrases générant des incohérences intra algorithme. Ainsi, des algorithmes différents attribuent des polarités différentes au même texte créant ainsi des incohérences inter-algorithmes. Dans ce papier, nous présentons une étude empirique détaillée afin de trouver les causes de ces incohérences et déterminer les facteurs d'influence. Notre étude porte sur 4 axes : une étude statistique pour qualifier les incohérences, une étude structurelle pour déterminer les liens entre la structure du texte et les incohérences, et une étude sémantique pour voir les liens entre la subjectivité du texte et les incohérences. Nous avons aussi étudié la relation entre les hyperparamètres dans les algorithmes d'apprentissage profond pour l'analyse de sentiment, la précision et l'incohérence dans ces algorithmes. Les travaux existants comme [1, 6, 11, 16], se sont intéressés à la génération des exemples contradictoires (adversarial exemples) pour tester la robustesse des algorithmes envers les incohérences. D'autres travaux comme celui de [11] se sont intéressés à la proposition d'algorithmes permettant aux outils d'analyses de sentiment d'éviter les incohérences intra-algorithmes.

Dans notre travail [8], nous nous intéressons à détecter les causes et les facteurs qui impactent les incohérences dans les outils d'analyse de sentiment. A notre connaissance, le travail présenté dans ce papier est le premier en soi qui propose une étude de la cohérences des outils d'analyses de sentiment couvrant plusieurs algorithmes et sur une large échelle de données.

Dans ce qui suit, nous présentons les contributions majeures ainsi que les résultats clés de notre étude.

2 CONTRIBUTIONS

2.1 Des algorithmes génériques pour les outils d'analyse de sentiment

Nous avons effectué, dans un premier temps une étude bibliographique des méthodes d'analyse de sentiment de la littérature, proposé des abstractions algorithmiques pour chaque catégorie d'algorithmes et sélectionné six algorithmes représentatifs à évaluer.

2.2 Évaluation Empirique

Nous avons mené une étude empirique qui qualifie les incohérences et explique la non robustesse des algorithmes d'analyse de sentiment en présence d'exemples contradictoires (adversarial exemples).

Notre évaluation porte sur deux types d'incohérences : des incohérences intra-algorithme qui se produisent dans le cas ou un algorithmes extrait des polarités différentes de deux phrases sémantiquement équivalentes, et des incohérences inter-algorithme qui se produisent lorsque deux algorithmes attribuent des polarités différentes à la même phrase. Nous avons évalué six algorithmes d'analyse de sentiment qui appartiennent à des catégories différentes pour définir les causes d'anomalies trouvées. Notre évaluation est effectuée sur quatre axes:

2.2.1 Étude Statistique. Le but de cette étude est de (1) vérifier si les incohérences sont fréquentes ou s'il s'agit d'anomalies rares, (2) trouver les différents types d'incohérences (3) déterminer les algorithmes et le type de données qui sont plus vulnérables à l'incohérence. Les résultats ont montré que les incohérences sont très fréquentes sur toutes les catégories d'algorithmes et dans tous les ensembles de données avec plus de présence dans les algorithmes basés sur l'apprentissage profond et sur les données factuelles.

2.2.2 Étude structurelle. Dans cette étude nous nous intéressons à la relation entre les incohérences et la structure des paraphrases évaluées. Nous nous sommes focalisés sur la relation entre les incohérences, la similarité sémantique et syntaxique entre les phrases sémantiquement équivalentes. Cela a permis de vérifier si les algorithmes sont vulnérables aux paraphrases avec une grande distance syntaxique. Les résultats ont montré que les algorithmes sont très sensibles à la différence syntaxique entre les textes sémantiquement équivalents.

2.2.3 Étude sémantique. Le but de cette étude est de vérifier si les incohérences dépendent de la subjectivité des paraphrases et si les incohérences dépendent de la précision des algorithmes. Autrement dit, nous vérifions si les algorithmes les plus précis présentent moins d'incohérences. Les résultats ont montré plus d'incohérences entre les données factuelles ainsi qu'une corrélation inverse entre les incohérences inter-algorithmes et la précision, i.e, plus l'algorithme est cohérent, plus il est plus précis.

2.2.4 Étude de l'effet des hyperparamètres sur la précision et les incohérences. Etant donné le constat effectué sur les outils basés sur l'apprentissage profond, nous avons étudié l'impact des hyperparamètres sur la précision et l'incohérence des algorithmes. Les résultats ont montré que les incohérences sont présentes sur toutes les configurations. En se basant sur leurs résultats, nous avons proposé des configurations qui permettent de minimiser les incohérences et maximiser la précision.

2.3 Proposition d'un benchmark de test

Afin d'évaluer des incohérences, nous avons construit un benchmark de paraphrases étiquetées avec des polarités. Le benchmark est construit à partir de cinq corpus de données pour l'analyse de sentiment disponibles publiquement que nous avons augmenté avec des paraphrases en utilisant la méthode [6]. La méthode que nous avons utilisée a une précision de 80%. Afin d'augmenter la qualité de notre benchmark, nous avons proposé une heuristique qui permet de réduire la marge d'erreur en gardant que les paraphrases valides. Cette heuristique permet de minimiser l'effort humain pour la vérification de la qualité de donnée et assure un benchmark de qualité avec un taux d'erreur réduit.

2.4 Recommandations.

En se basant sur les résultats d'évaluation, nous proposons un ensemble de recommandation pour choisir des algorithmes d'analyse de sentiment adapté à un scénario donné et avec différents types de données.

REFERENCES

[1] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998* (2018).
[2] Erik Cambria, Soujanya Poria, Devamanyu Hazarika, and Kenneth Kwok. 2018. SenticNet 5: discovering conceptual primitives for sentiment analysis by means of context embeddings. In *Proceedings of AAAI*.
[3] Nicholas Carlini and David Wagner. 2017. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 3–14.
[4] Haibo Ding and Ellen Riloff. 2018. Weakly supervised induction of affective events by optimizing semantic consistency. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
[5] Guohong Fu, Yu He, Jiaying Song, and Chaoyue Wang. 2014. Improving Chinese sentence polarity classification via opinion paraphrasing. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*. 35–42.
[6] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial Example Generation with Syntactically Controlled Paraphrase Networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 1875–1885.
[7] Xiang Ji. 2014. Social data integration and analytics for health intelligence. In *Proceedings VLDB PhD Workshop*.
[8] Wissam Mammar Kouadri, Mourad Ouziri, Salima Benbernou, Karima Echihabi, Themis Palpanas, and Iheb Ben Amor. 2020. Quality of sentiment analysis tools: the reasons of inconsistency. *Proceedings of the VLDB Endowment* 14, 4 (2020), 668–681.
[9] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006* (2017).
[10] Taylor Mahler, Willy Cheung, Micha Elsner, David King, Marie-Catherine de Marneffe, Cory Shain, Symon Stevens-Guille, and Michael White. 2017. Breaking nlp: Using morphosyntax, semantics, pragmatics and world knowledge to fool sentiment analysis systems. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*. 33–39.
[11] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725* (2016).
[12] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 856–865.
[13] Julian Risch and Ralf Krestel. 2018. Aggression identification using deep learning and data augmentation. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. 150–158.
[14] Huan Rong, Victor S Sheng, Tinghuai Ma, Yang Zhou, and Mznah A Al-Rodhaan. 2020. A Self-play and Sentiment-Emphasized Comment Integration Framework Based on Deep Q-Learning in a Crowdsourcing Scenario. *IEEE Transactions on Knowledge and Data Engineering* (2020).
[15] Soroush Vosoughi, Prashanth Vijayaraghavan, and Deb Roy. 2016. Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 1041–1044.
[16] Ke Wang and Xiaojun Wan. 2018. SentiGAN: Generating Sentimental Texts via Mixture Adversarial Networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. 4446–4452.
[17] Jason W Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196* (2019).

Architectures Transformeurs pour la classification multilabels de textes

Haytame Fallah
Aix Marseille Univ, Université de Toulon, CNRS, LIS
Marseille, France
Hyperbios
Toulon, France
haytame.fallah@lis-lab.fr

Patrice Bellot
Aix Marseille Univ, Université de Toulon, CNRS, LIS
Marseille, France
patrice.bellot@univ-amu.fr

Emmanuel Bruno
Université de Toulon, Aix Marseille Univ, CNRS, LIS
Toulon, France
emmanuel.bruno@univ-tln.fr

Elisabeth Murisasco
Université de Toulon, Aix Marseille Univ, CNRS, LIS
Toulon, France
elisabeth.murisasco@univ-tln.fr

ABSTRACT

Les modèles de langue pré-entraînés ont prouvé leur efficacité dans la classification de texte multiclasses. Notre objectif est d'étudier et d'améliorer ce type d'approches pour la classification multilabels de texte, une tâche étonnamment peu explorée au cours de ces toutes dernières années. Cette tâche a pourtant des applications industrielles importantes telles que la recommandation de contenu, l'extraction de méta-données pour l'enrichissement des bases de données ou le routage automatique multicritères des emails. Dans cet article, notre originalité est de proposer des méthodes d'exploitation des activations des couches de sortie des transformeurs pour améliorer la performance de ces modèles pour la classification multilabels. Notre contribution concerne l'évaluation de l'utilité des méthodes de seuillage sur plusieurs modèles d'apprentissage profond, en calculant un seuil de classification global pour optimiser l'ensemble des classes *(SGO)*, ou un seuil individuel propre à chaque classe étudiée *(SI)*. Elle concerne aussi la proposition de deux approches pour la classification multilabels de texte. La première approche (*NPA*) consiste à ajouter un paramètre pour l'apprentissage du nombre de classes et/ou labels N présentes pour un exemple donné, pour considérer les classes qui correspondent aux N activations les plus élevées comme étant des labels valides. La deuxième approche *(TL)* consiste à ajouter une couche au transformeur pour l'apprentissage des critères utiles pour la sélection des labels pertinents. Nous évaluons ces approches sur des corpus d'articles de journaux et d'articles scientifiques. Nous avons aussi constitué et mis à disposition un jeu de données de résumés d'articles scientifiques en français que nous avons conçu à partir du dépôt d'archives ouvertes *'HAL'*. Ces évaluations montrent que la performance de nos propositions dépasse celles des méthodes de l'état de l'art de classification multilabels de texte pour les jeux de données étudiés, et sont transposables à tout problème de classification multilabels utilisant les réseaux de neurones.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; **Natural language processing**.

KEYWORDS

Apprentissage profond, Classification automatique, Multilabels, Modèles de langue, Transformeurs, BERT.

1 INTRODUCTION

Les modèles de langue probabilistes sont depuis longtemps utilisés pour de très nombreuses tâches du traitement automatique du langage naturel et de la recherche d'information. Nous nous intéressons à l'adaptation de ces modèles pour la classification multilabels où différentes parties du texte contribuent différemment pour la prédiction des labels.

L'adaptation de l'apprentissage profond pour la modélisation séquentielle du langage a débuté avec l'utilisation des réseaux de neurones récurrents (RNN). L'entraînement de ces réseaux se fait d'une manière itérative où un état caché h est mis à jour à partir de chaque nouveau mot d'une séquence d'entrée. Les informations contenues dans chaque composante de la phrase sont ainsi préservées tout au long du traitement de celle-ci, mais avec une contribution qui diminue proportionnellement à la longueur de la séquence.

Les réseaux récurrents de longue mémoire à court terme (LSTM) [Hochreiter and Schmidhuber 1997] ont été conçus pour remédier à ce problème en ajoutant, en plus des états cachés h , des cellules de mémoire qui permettent de réguler le flux d'information et de retenir ou "oublier" les éléments des étapes antérieures selon leur importance. Ce mécanisme a des limitations car l'importance donnée à un mot n'est pas relative au reste de la phrase.

Le mécanisme d'attention [Bahdanau et al. 2016] a été ensuite introduit dans l'utilisation des RNNs. Il permet de donner une importance (poids) relative à chaque mot de la séquence calculée à l'aide d'un réseau de neurones feed-forward (propagation avant). Ces modèles permettent une bonne modélisation du langage et réussissent à capturer la nature séquentielle des phrases, mais avec un temps d'entraînement relativement long. En n'utilisant que le mécanisme d'attention, les transformeurs [Vaswani et al. 2017] peuvent être entraînés en parallélisant le traitement d'une séquence tout en gardant l'information sur l'ordre des mots. Ces transformeurs disposent d'une partie "encodeur" et une autre "décodeur" [Sutskever et al. 2014], composée chacune de plusieurs couches. Les plongements de mots (embeddings), l'entrée de ces modèles, passent par

des couches d'attention (ou self-attention) où ils sont mis à jour en fonction des scores d'attention calculés à partir des autres parties de la séquence, et cela pour avoir une représentation contextualisée pour chaque composante des phrases.

Les transformeurs sont actuellement les meilleures implantations des modèles de langue. Le modèle BERT, pour Bidirectional Encoder Representations from Transformers [Devlin et al. 2019], et ses variantes sont parmi les plus récents. BERT se différencie des autres transformeurs par sa capacité à traiter la séquence de texte d'une manière bi-directionnelle à l'opposé de GPT-2 [Radford et al. 2019] qui ne regarde que la partie gauche du mot en cours de traitement, ou ELMO [Peters et al. 2018] qui concatène la représentation gauche et droite de la séquence, ce qui rend l'apprentissage plus lent. Cette bi-directionnalité de l'apprentissage améliore les représentations linguistiques du modèle car le sens d'un mot de la phrase peut dépendre non seulement des mots qui le précèdent mais aussi de ceux qui le suivent. Ces modèles sont très utiles compte tenu de leur ré-utilisabilité, en pré-entraînant un modèle en non-supervisé sur un large corpus de texte, puis en l'adaptant pour la tâche de traitement de texte souhaitée. Le mécanisme d'attention peut être très performant pour capturer l'importance de certaines parties de l'entrée par rapport à d'autres.

La classification multilabels de texte est une tâche de traitement automatique du langage où un texte en entrée peut être associé à plusieurs classes. Le but étant de pouvoir extraire tous les sujets, prédéfinis ou non, contenus dans ce texte. Plusieurs approches d'apprentissage machine ont été proposées pour aborder ce problème. À notre connaissance, aucun benchmark officiel n'existe pour la classification multilabels de texte, et cela en dépit du fait que cette tâche ait des applications concrètes et importantes dans le monde industriel. Parmi ces applications on retrouve l'indexation multicritères, la recommandation de contenu, ou même l'extraction des méta-données pour l'enrichissement des bases de données applicable sur divers types de corpus, à l'instar des articles scientifiques. Les travaux présentés dans cet article sont réalisés dans le cadre d'une thèse CIFRE en collaboration avec Hyperbios, une société de services informatiques, pour la classification et la segmentation des demandes clients exprimées par mail pour le compte de plusieurs agences d'assurance. La classification multilabels permettra d'identifier les sujets contenus dans chaque mail, et déclencher les actions correspondantes par les systèmes d'automatisation de tâches et de proposition de réponses.
Exemple :
Madame, Suite à notre conversation téléphonique je vous adresse en pièces jointes la photo du compteur kilométrique du véhicule xxxx ainsi que la photo de la plaque d'immatriculation. J'ai bien noté que le véhicule pouvait être conduit par un tiers, et qu'en cas de sinistre il n'y aurait pas de franchise. Je vous remercie de bien vouloir m'adresser aussi rapidement l'attestation d'assurance.
Classifications : Envoi de documents, informations relevé compteur, demande d'attestation.
Cette exemple illustre la complexité du problème qui consiste à extraire tous les labels possibles depuis un texte de taille réduite, ayant peu d'éléments contextuels et sémantiques. Les différents styles d'écritures ainsi que l'emploi de termes qui diffèrent selon le niveau d'expertise du client pour désigner le même concept rajoutent des contraintes supplémentaires à cette problématique.

Dans cet article, nos contributions concernent essentiellement :

- La création d'un corpus de données de classification de texte multilabels en français, contenant les résumés d'articles scientifiques obtenus à partir de l'archive ouverte "*HAL*";
- L'évaluation des performances des modèles de langue disponibles : BERT et ses variantes DistilBERT, RoBERTa et DeBERTa pré-entraînés principalement sur des corpus en anglais, ainsi que CamemBERT et FlauBERT, deux autres variantes de BERT pré-entraînées sur des corpus en français;
- L'étude des méthodes de sélection de seuil pour une exploitation plus efficace des résultats de ces modèles, notamment le choix d'un seuil global qui optimise les performances de toutes les classes, ou le calcul d'un seuil propre à chaque classe pour l'optimisation individuelle des labels;
- La proposition de deux approches alternatives au seuillage pour la sélection des classes pertinentes. La première consiste à introduire un paramètre à la dernière couche du transformeur qui sera entraîné pour le calcul du nombre de classes présentes dans un exemple, la valeur de ce paramètre sera utilisée pour sélectionner les labels ayant les activations les plus fortes; La deuxième consiste à rajouter une couche finale au modèle, qui aura le même nombre de paramètres que l'avant dernière couche (égal au nombre de classes), dans le but d'obtenir des valeurs d'activation plus discriminantes pour l'exemple donné, ie. activation élevée si présence de label, faible dans le cas contraire.

Nous évaluons ces approches sur des corpus comparables d'articles scientifiques, HAL-Dataset pour le français et AAPD pour l'anglais, mais aussi sur des articles d'actualité en anglais (corpus de Reuters). Les modèles et les architectures proposés sont comparés à des approches de référence, performantes et plus transparentes sur les critères de classification appris, comme les arbres de décision et les machines à vecteur support (SVM).

L'article est organisé de la façon suivante : la section 2 présente les approches qui abordent la classification multilabels, les sections 3 et 4 décrivent la méthodologie suivie et les approches proposées et la section 5 est dédiée aux expérimentations.

2 TRAVAUX ANTÉRIEURS

Dans la classification multiclasses, chaque exemple (instance) X du jeu de données est associé à un label unique. La classification multilabels (CMLT) consiste en plus à pouvoir associer chaque entrée avec plusieurs labels $\mathbf{Y}$, plutôt qu'un seul.

2.1 Méthodes de classification multilabels

Les méthodes de classification multilabels peuvent être classées en trois catégories principales : par transformation du problème, par adaptation ou par des méthodes d'ensembles.

2.1.1 Transformation du Problème (PT). La transformation du problème consiste à 'transformer' le jeu de données pour changer le problème en une classification multiclasses à label unique. Une de ces méthodes consiste à considérer toutes les combinaisons uniques de labels possibles du jeu de données, *label powerset* [Tsoumakas et al. 2010], et à entraîner un classifieur multiclasses $M : X \rightarrow P(Y)$, où

$P(Y)$ est le powerset de **Y**, l'ensemble des sous-ensembles uniques et distincts de labels.

En plus du nombre élevé de classes possibles qui peut atteindre $2^{|Y|}$, le challenge réside dans la capacité à trouver suffisamment d'exemples pour chaque combinaison de labels pour éviter la sous-représentation des classes.

La pertinence binaire [Boutell et al. 2004], est une autre méthode de transformation de problème où sont entraînés |**Y**| classifieurs binaires qui détectent la présence ou la non-présence d'un label pour une instance. |**Y**| jeux de données sont construits à partir du jeu original. Chaque jeu de données D_y, $y \in \mathbf{Y}$, contient les instances où y est un label qui les caractérise. Pour une instance x du jeu de données, le résultat de la classification est l'union des labels détectés par chaque classifieur.

Pour un |**Y**| grand, le temps d'entraînement et d'inférence des modèles est élevé. Il est important de noter, qu'en transformant le problème en une classification multiclasses, les dépendances qui peuvent exister entre les différents labels ne sont plus considérées [Luaces et al. 2012].

2.1.2 Méthodes d'ensembles. Un ensemble de classifieurs multiclasses peut être combiné pour créer un classifieur multilabels. Pour une instance donnée, chaque classifieur va prédire une seule classe et toutes les sorties de ces classifieurs sont alors combinées via une méthode d'ensemble. Une de ces méthodes consiste à considérer une classe comme présente si un pourcentage de classifieurs ayant prédit cette classe est atteint, aussi appelé seuil discriminatif [1]. L'algorithme *RAKEL* [Tsoumakas and Vlahavas 2007] est une autre variation de cette méthode. Des classifieurs entraînés sur des sous-ensembles aléatoires des *labels powersets* sont utilisés pour la création d'un classifieur multilabels, les prédictions de ces classifieurs passent par un système de vote pour la prédiction finale.

L'utilisation de plusieurs classifieurs impose des contraintes fortes en termes d'espace mémoire, ainsi que la nécessité d'optimiser un nombre de modèles qui augmente linéairement avec le nombre de classes du jeu de données.

2.1.3 Adaptation du Problème (PA). Les méthodes d'adaptation du problème ne nécessitent pas une transformation du jeu de données mais une adaptation des algorithmes de classification, tels que ML-kNN [Zhang and Zhou 2007] qui étend l'algorithme kNN pour les données multilabels, ou BP-MLL [Min-Ling Zhang and Zhi-Hua Zhou 2006] une adaptation de l'algorithme de rétro-propagation pour les réseaux de neurones.

L'adaptation des algorithmes d'apprentissage profond pour le multilabels reste de manière générale une voie avec peu de contributions. Une adaptation de ces approches pourraient contribuer à une augmentation significative des performances pour la tâche de classification multilabels. L'utilisation d'un seul modèle sans le recours à une transformation préalable des données constitue une méthode efficiente pour essayer de répondre au problème du multilabels.

2.2 Approches de seuillage

Plusieurs méthodes de seuillage ont été proposées pour les approches citées précédemment, ce sont des méthodes qui impactent directement le choix d'une classe pour le problème multilabels. Le seuil peut être ajusté de plusieurs façons, soit pour optimiser toutes les classes (un seuil global), ou pour optimiser individuellement chaque classe (nombre de seuils égal au nombre de classes). Soit m le nombre d'exemples dans le jeu de données de test (ou la validation) et n_y le nombre de classes (labels). Les quatre stratégies les plus utilisées pour le choix du(des) seuil(s) sont :

- **SCut**: Les classes sont optimisées de façon individuelle, les seuils sont choisis en fonction des performances sur le jeu de données de validation, mesurés soit par la maximisation d'un score ou la minimisation d'une fonction de coût, et sans garantir l'obtention d'un optimum global. Cette méthode peut aussi être utilisée pour obtenir un seuil global pour toutes les classes [Yang 2001];
- **RCut** (Rank Cut) : pour chaque instance, les classes sont ordonnées selon le score obtenu, les t premières classes sont choisies comme labels pertinents. Le paramètre t peut être fixé ou réglé à partir du jeu de données de validation [Yang 2001];
- **PCut** (Proportion Cut) : pour chaque classe y_i, les instances du jeu de données de test sont ordonnées selon le score obtenu pour cette classe. Les k_i premières instances sont choisies pour la classe y_i où $k_i = P(y_i) \times x \times n_y$ est le nombre d'instances attribuées à cette classe. $P(y_i)$ est la probabilité qu'une instance fasse partie de la catégorie y_i (calculée préalablement à partir du jeu d'entraînement), et x le nombre moyen d'instance à attribuer pour une classe quelconque fixé au préalable. Si $x = n$ toutes les instances sont prises pour une catégorie, pour $x = 0$ aucun exemple n'est considéré comme faisant partie de la classe étudiée [Lewis et al. 1996; Yang 1997];
- **MCut** (Maximum Cut) : les labels sont ordonnés à partir des scores obtenus pour une instance du jeu de données, le seuil est égal à la moyenne des deux labels contigus pour lesquels l'écart de score est le plus important [**?**].

Des variations de ces méthodes visant à améliorer leur performance et pallier aux problèmes qu'elles peuvent poser ont été suggérées [Yang 2001]. Les méthodes axées sur les scores de classification sont les meilleures parmi les approches de seuillage [**?**]. Nous choisissons dans cet article la méthode SCut pour l'évaluation des modèles de langues étudiés.

2.3 Utilisation de l'apprentissage profond

Les modèles d'apprentissage profond ont aussi été utilisés pour répondre au problème de la classification de texte dont les CNN [Kim 2014], RCNN [Lai et al. 2015] et HAN [Yang et al. 2016] mais sans se focaliser sur le problème multilabels.

[Gan et al. 2019] utilise un réseau de neurones pour la classification multilabels dans le cadre de l'analyse multi-composition spectroscopique, un réseau composé d'une partie classificateur à laquelle un paramètre est ajouté pour l'apprentissage du seuil d'activation optimal pour la classification. Ce paramètre sera optimisé en fonction du seuil calculé en appliquant le modèle en cours d'apprentissage sur le jeu de données d'entraînement, la valeur cible du seuil sera donc différente pour chaque itération de la phase

[1] https://www.scikit-yb.org/en/latest/api/classifier/threshold.html

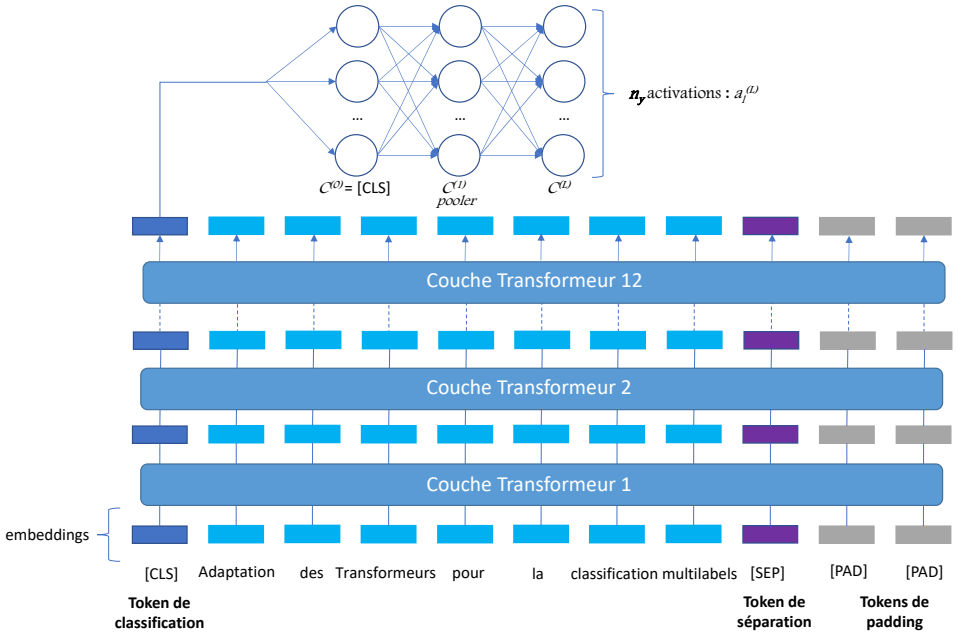


Figure 1: Architecture BERT avec une couche dense de classification au-dessus des couches transformeurs et les valeurs d'activation des sorties.

d'entraînement, ce qui peut engendrer une difficulté de convergence de la fonction d'erreur vers son minimum.

Par ailleurs, MAGNET [Pal et al. 2020] utilise les plongements de BERT [Devlin et al. 2019] comme entrée, il concerne explicitement la CMLT et réussit à avoir de bonnes performances en score F1 [2] pour les jeux de données d'AAPD et de Reuters (cf. section 5.2). DocBERT [Adhikari et al. 2019], qui est maintenant une référence de l'état de l'art, rajoute un réseau linéaire à la tête du transformeur BERT, mais sans traitement par la suite des sorties du modèle.

Les transformeurs ont été utilisés pour la classification multilabels de texte dite 'extrême' [Chang et al. 2019; Gong et al. 2020] où sont traités des corpus très larges de textes ayant un nombre de labels pouvant atteindre les dizaines de milliers. De telles architectures ne sont pas adaptés à la problématique de textes courts.

Les tentatives d'utiliser les réseaux de neurones pour la classification de texte ne se focalisent pas sur la classification multilabels. Celles qui traitent ce problème ne donnent généralement pas d'importance à la manière dont sont exploitées les activations de la couche de sortie (e.g utilisation de seuil) ou en modifiant l'architecture du réseau pour essayer de l'adapter à cette tâche.

3 ADAPTATION DES MODÈLES DE LANGUE PRÉ-ENTRAÎNÉS

Nous nous intéressons dans cet article à l'adaptation des modèles transformeurs, notamment BERT et ses variantes, pour la classification multilabels de texte.

3.1 Adaptation de BERT pour la classification multilabels

Pour les réseaux de neurones, le texte d'entrée doit être segmenté et converti en une liste de symboles (appelés tokens) pour le traitement. Ces tokens sont par la suite associés à des identifiants uniques, définis dans le dictionnaire utilisé par le modèle, et aux plongements (word embeddings) de mots qui leur correspondent. Ces derniers constituent les données de la couche d'entrée de ces modèles. L'architecture des modèles de langue à base de transformeurs se différencie de leurs prédécesseurs par leur capacité à paralléliser le processus d'entraînement en traitant tout le vecteur (tous les tokens) de l'entrée simultanément. L'entraînement est réalisé en masquant des tokens (de façon aléatoire ou non) et en essayant ensuite de les prédire en utilisant principalement le mécanisme d'attention, processus appelé modélisation masquée du langage.

Le modèle BERT introduit la bi-directionnalité dans la prédiction des tokens masqués, où les deux contextes sémantiques gauche et droit du mot à prédire sont pris en compte. En plus de la modélisation masquée du langage, BERT est entraîné pour la prédiction de la phrase suivante, une tâche où le modèle reçoit une paire de phrases et essaie de prédire si la deuxième phrase de la paire suit la première dans le texte original.

BERT introduit aussi un token spécial de classification **[CLS]** (possédant aussi un identifiant et un vecteur d'embedding) contenant un *état caché* de la phrase, mis à jour dans chacune des couches du modèle. Un réseau linéaire de L couches denses (généralement L=2) constitue les dernières couches du modèle avec le token **[CLS]** comme entrée, et n_y sorties (approche similaire à [Adhikari et al. 2019]. La figure 1 présente l'architecture du modèle.

[2] $F1 - score = 2 \times \frac{précision \times rappel}{précision\ rappel}$

Le choix des fonctions d'activation et des fonctions de coût dépend de la nature de la tâche de classification. Dans le cas de la classification multiclasses où les classes sont mutuellement exclusives (une instance doit appartenir à une et une seule classe) on utilise classiquement la fonction *Softmax* comme fonction d'activation de la dernière couche du modèle ($C^{[L]}$). Elle a pour but de convertir le vecteur de sorties, les activations de $C^{[L]}$, en un vecteur de probabilités proportionnelles aux valeurs de ces activations. L'entropie croisée (Cross Entropy) est utilisée comme fonction d'erreur du modèle.

Dans le cas de la classification multilabels, la nature de l'architecture des réseaux de neurones peut être exploitée de telle manière à utiliser les valeurs des activations $A^{[L]}$ de la couche de sorties $C^{[L]}$ pour déterminer la présence ou non d'un label, et cela en introduisant un seuil. Ce seuil, s'il est dépassé, permettra de considérer ou non le label étudié. Le but n'étant pas d'avoir une distribution de probabilités en fonction de toutes les classes, la fonction *Softmax* n'est pas adaptée. La fonction d'activation *Sigmoid* σ est plus appropriée pour cette tâche. Elle convertit chaque activation de $A^{[L]}$ en une valeur comprise entre 0 et 1, calcule une probabilité de présence de chaque label y_i en fonction des valeurs de l'activation qui lui correspond $a_{y_i}^{[L]}$ de la couche L. Cela implique l'utilisation de l'entropie croisée binaire (Binary Cross Entropy - BCE) comme fonction de coût qui vise à réduire l'écart entre $a_{y_i}^{[L]}$ et la vraie valeur de sortie (0 ou 1 selon la présence ou non du label y).

Cette approche ne peut être considérée complètement comme une transformation du problème car la méthode ne requiert pas une transformation du jeu de données ou la création de multiples classifieurs binaires, ni une adaptation complète de l'apprentissage profond pour la classification multilabels car les résultats du réseau de neurones doivent être traités en aval pour avoir les classifications finales.

3.2 Apprentissage profond et méthodes de seuillage

Les approches de seuillage peuvent être appliquées au transformeur, si l'on considère que chaque activation de la couche finale est un classifieur binaire de la classe qu'il représente.

3.2.1 Seuil global de classification (SGO). Le seuil de classification s peut être choisi de façon à maximiser l'ensemble des scores de classification. Pendant l'entraînement du modèle et après chaque itération, la valeur de s est variée de 0 à 1, avec un pas défini au préalable (10^{-2}). Le score micro-F1 est ensuite calculé pour chaque seuil s. On obtient au final le seuil global optimal s_{go} qui permet d'obtenir les meilleures performances sur le jeu de données d'entraînement. Ce seuil est ensuite utilisé pour les jeux de données de validation et de test. L'ensemble des labels L présents pour un exemple x peut être exprimé sous la forme:

$$Y_{x \in X} = \cup_{y \in Y} \{y_i\} : \sigma(a_{y_i}^{[L])}) \geq s_{go}$$

$a_{y_i}^{[L]}$ étant l'activation qui correspond au label y_i parmi les activations $A^{[L]}$.

Une autre variation la méthode SCut (avec seuil global) consiste à affiner le seuil optimal à partir du jeu de données de la validation pour ensuite l'appliquer au jeu de données de test. Seule la première approche a été étudiée dans cet article.

3.2.2 Seuils individuels (SI). L'utilisation d'un seuil s commun pour toutes les classes suppose que les descripteurs et les poids directement liés aux activations $A^{[L]}$ soient les mêmes pour chaque classe/label. Ce qui n'est pas le cas, l'intensité d'activation d'un neurone de la dernière couche $a_{y_i}^{[L]}$, lors de la présence du label y pour une instance donnée, varie d'une classe à l'autre, un effet qui s'accentue si le jeu de données est peu équilibré.

Nous proposons donc d'évaluer la méthode SCut (avec seuils individuels) en mettant en place un seuil s_y pour chaque label y du jeu de données, attribué aux activations $a_{y_i}^{[L]}$ de $C^{[L]}$. Les valeurs de s_y sont les valeurs qui maximisent les scores de classification pour un label y. Ces seuils sont calculés comme pour le s_{go} pendant la phase d'entraînement du modèle, et sur le jeu de données d'entraînement, en faisant varier chaque seuil de façon à maximiser le score F1 de chaque label.

$$Y_{x \in X} = \cup_{y \in Y} \{y_i\} : \sigma(a_{y_i}^{[L]}) \geq s_y$$

Comme pour le seuil global, les seuils individuels peuvent être calculés à partir du jeu de données de validation.

4 APPROCHES PROPOSÉES

Nous proposons deux méthodes alternatives au seuillage pour une exploitation plus efficace des valeurs des activations $A^{[L]}$ de la couche de sortie. Ces approches ont pour but de s'affranchir d'un calcul de seuil en apprenant des caractéristiques propres au texte permettant la bonne sélection des labels pertinents.

4.1 N plus grandes activations (NPA)

L'utilisation d'un seuil pour déterminer la présence ou non d'un label entraîne dans plusieurs cas une sous classification (respectivement sur classification) d'une instance quand le nombre prédit de labels est inférieur (respectivement supérieur) au nombre effectif de labels présents. La première alternative proposée consiste à introduire un paramètre (neurone) à la dernière couche de classification et qui sera utilisé uniquement pour le calcul du nombre de labels y présents pour une instance. Soit $A'^{[L]}$ la liste des N plus grandes activations pour un exemple donné, N étant le nombre effectif des labels présents pour cet exemple :

$$Y_{x \in X} = \cup_{y \in Y} \{y_i\} : a_{y_i}^{[L])} \in A'^{[L])}$$

L'optimisation de ce paramètre se fera en utilisant le nombre de classes présentes pour une instance comme valeur cible, et une fonction d'erreur adaptée au problème de régression (par ex. l'erreur absolue moyenne **MAE** ou l'erreur quadratique moyenne **MSE**).

Nous utilisons un optimiseur unique pour la propagation arrière (calculs des gradients des fonctions d'erreurs et mise à jour des poids du modèle), par conséquent, l'erreur de la régression doit être mise à la même échelle que l'erreur de la classification, ceci est fait en réduisant l'erreur de régression par un facteur de 5. La valeur de ce paramètre sera utilisée pour récupérer les N plus grandes activations $a_{y_i}^{[L]}$ qui seront considérées comme labels prédits. La figure 2 présente l'architecture du modèle pour cette approche.

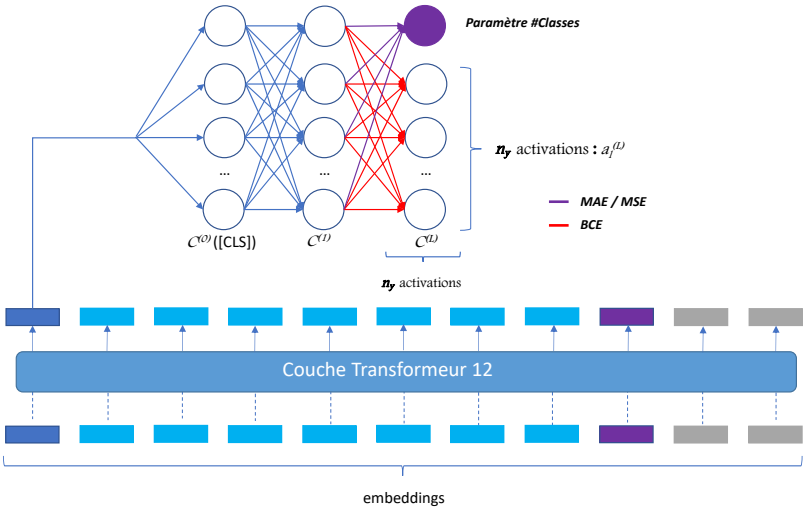


Figure 2: Architecture couche de classification dense avec un paramètre pour le calcul du nombre de classes présentes.

Le token [CLS], la sortie finale des couches transformeurs, contient une représentation du texte riche en informations. L'objectif de cette architecture est de pouvoir extraire à partir de ce token, et de l'état caché de la phrase qu'il contient, des critères ou des informations sur le nombre de sujets distincts présents dans le texte. Des critères qui serviront au calcul du paramètre.

4.2 Couche de seuillage (Threshold Layer TL)

La deuxième approche proposée se caractérise par l'ajout d'une couche dense à l'aval du classifieur, pour un total de $L = 3$ couches, ayant n_y neurones pour correspondre à la dernière couche de classification du modèle. Cet ajout a pour but de faire rapprocher les valeurs des activations finales le plus possible de 1 dans le cas de la présence du label, ou d'une valeur nulle dans le cas contraire. Par conséquent, le seuil trivial de $0,5$ peut être utilisé comme seuil pour la classification.

L'ajout de cette couche pourrait avoir comme effet l'augmentation du rappel du classifieur car plusieurs activations des labels non présents ne dépasseront plus le seuil de classification défini du fait qu'elle se rapprocheront le plus possible de 0. Un gain en précision peut aussi être espéré du fait que les activations des labels présents seront renforcées et mises plus en avant par rapport aux activations des labels non pertinents.

Pour cette approche, nous utilisons deux optimiseurs, un premier qui englobe et optimise toutes les couches du transformeurs ainsi que les deux premières couches du classifieur, et un autre dédié à l'optimisation de la dernière couche que l'on a rajoutée. La fonction d'erreur utilisée est la même pour les deux parties du classifieur, la BCE en l'occurrence. L'architecture finale du classifieur est présentée dans la figure 3

5 EXPÉRIMENTATIONS

Nous présentons dans cette section les résultats de l'évaluation de plusieurs modèles de langue transformeurs, principalement BERT et ses variantes, sur trois jeux de données de textes multilabels, dont le jeu de données français "HAL-Dataset" que nous avons conçu et mis à disposition. Nous comparons les différentes méthodes citées, ie. les méthodes de seuillage ainsi que les alternatives proposées à des approches de références, le tout mis en perspective avec des résultats cibles optimaux (approches oracles).

5.1 Modèles de langue évalués

Pour l'évaluation des méthodes proposées, nous utilisons la version *base* de **BERT**, avec 12 couches transformeurs et une taille du vecteur d'embeddings de 768, ainsi que quelques-unes de ses variantes :

- **RoBERTa** [Liu et al. 2019] une variante avec un processus d'entraînement optimisé, caractérisé par la suppression de la partie *prédiction de la phrase suivante* (ou *Next Sentence Prediction*) de BERT. Le jeu de données utilisé pour l'entraînement de RoBERTa est dix fois plus grand que celui utilisé pour BERT. Ceci a contribué à un gain en performance de RoBERTa par rapport à la version originale sur les tâches de traitement automatique du langage dans le comparatif GLUE [3];
- **DistilBERT**, une autre variante de BERT optimisée par l'utilisation de la distillation des connaissances pour avoir une approximation de BERT avec 40% moins de paramètres, tout en gardant 97% de ses performances. L'idée pour DistilBERT est basée sur le fait qu'après l'entraînement d'un modèle très grand, la distribution de la sortie peut être approximée par un réseau de neurones beaucoup plus petit, en utilisant la divergence Kulback-Leiber [Kullback and Leibler 1951] comme fonction d'optimisation [Sanh et al. 2020];
- **DeBERTa**, la variante la plus récente de BERT où les mots sont représentés par deux vecteurs qui encodent leur contenu et leur position relative dans la phrase. Il est aussi caractérisé

[3] https://gluebenchmark.com/tasks

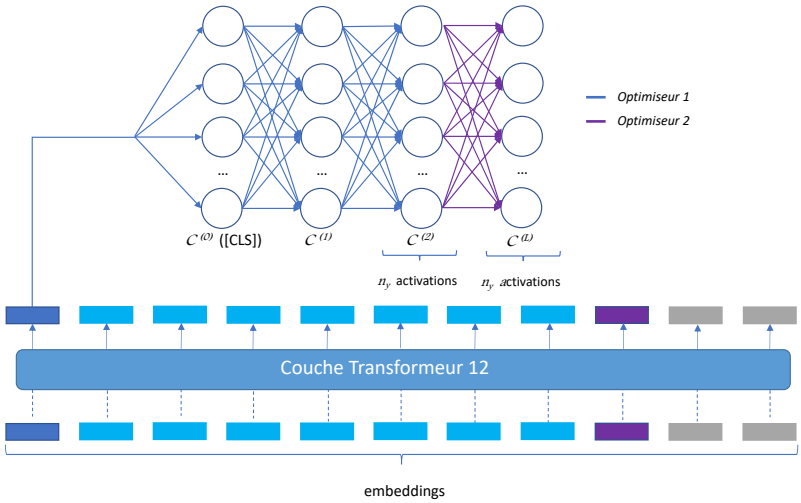


Figure 3: Architecture couche de classification dense de seuillage rajoutée à la fin du classifieur.

par l'optimisation du décodage de la prédiction des tokens masqués, ce qui contribue à un gain significatif en efficience dans la phase de pré-entraînement du modèle, mais aussi dans les performances concernant les différentes tâches de traitement du langage naturel. [He et al. 2021],
- **CamemBERT** est une variante basée sur RoBERTa [Martin et al. 2020], entraînée sur la partie française du corpus OSCAR [Suárez et al. 2019];
- **FlauBERT** [Le et al. 2020] variante entraînée sur divers sous-corpus français de différents styles d'écritures, allant des écritures formelles (par ex. Wikipedia et livres), aux écrits extraits d'internet (par ex. Common Crawl).

La comparaison des différents modèles de langue nous semble intéressante pour pouvoir évaluer les performances de nos approches ainsi que leur réutilisabilité et leur transposabilité sur différentes architectures de réseaux de neurones.

5.2 Corpus de textes utilisés

Comme nous l'avons déjà signalé, aucun LeaderBoard dédié à cette tâche n'existe. On retrouve peu de corpus multilabels utilisés d'une manière fréquente dans la littérature pouvant servir à l'évaluation et la comparaison des modèles, et ceci d'autant plus pour la langue française. Pour y remédier, nous avons constitué notre propre corpus de résumés d'articles scientifiques écrits en français à partir de HAL, corpus que nous mettons à disposition de tous.

Dans cette section, nous fournissons des détails concernant les différents jeu de données utilisés [4] pour l'évaluation des différents modèles :

- **HAL-Dataset** est un jeu de données en français que nous avons extrait de la plateforme d'archives ouvertes "HAL". Il contient les résumés de publications scientifiques de trois domaines (mathématiques, physique et informatique), publiées entre 1950 et 2020. L'extraction de ces résumés a été faite grâce à l'outil mis en disposition par HAL [5], en ne gardant que les articles pour lesquels le champs "résumé" français est présent. Ce qui représente 12 430 documents répartis sur 194 classes différentes, ici découpés en jeux d'entraînement, de validation et de test;
- **Reuter-21578** [6] est une collection d'articles du fil d'actualités de Reuters de l'année 1987 et publiée à l'année 1990. C'est un jeu de données qui a beaucoup été utilisé pour évaluer les modèles pour la CMLT de textes. Un article peut appartenir à un ou plusieurs domaines parmi 90;
- **AAPD** (ou ArXiv Academic Paper Dataset) est, comme pour le jeu de données "HAL-Dataset", une collection de la section "Résumé" (abstract) de plusieurs publications scientifiques. Un article scientifique peut avoir une ou plusieurs classifications parmi 54 classes. On utilise la même répartition entraînement, validation et test que [Yang et al. 2018].

Le tableau 1) et la figure 4 présentent plus en détail les différentes caractéristiques de ces jeux de données.

5.3 Méthode d'évaluation

Nous allons comparer les deux approches proposées ainsi que l'application des méthodes de seuillage aux transformeurs à des méthodes plus transparentes sur les critères de sélection, mais aussi avec des adaptations des modèles d'apprentissage profond à la classification multilabels de texte. Nous proposons aussi de comparer l'ensemble de ces méthodes à plusieurs bornes supérieures qui représentent les résultats optimaux que l'on peut atteindre.

5.3.1 Approches de références. Nous proposons dans un premier temps une comparaison avec des approches non-neuronales, avec

[4] Tous les jeux de données sont téléchargeables ici : https://github.com/hf-lis/Multi-Label-Datasets

[5] API HAL : https://api.archives-ouvertes.fr/docs/search

[6] https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection

Table 1: Les jeux de données avec W le nombre moyen de mots par document.

	#Train	#Valid	#Test	labels	W	#Train Epochs
Reuter-21578	5827	1943	3019	90	127,76	120
AAPD	53840	1000	1000	54	163,16	40
HAL-Dataset	9944	1243	1243	194	112,72	50

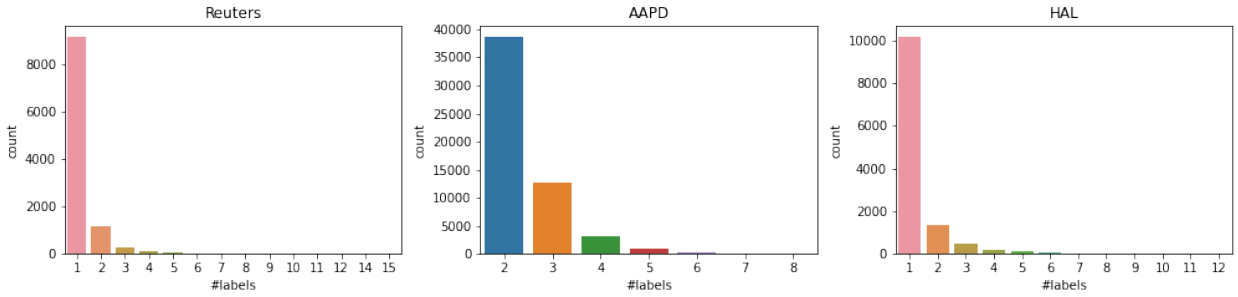


Figure 4: Compte des instances en fonction du nombre de labels pour l'ensemble des corpus.

des critères de classification interprétables, d'apprentissage machine, en utilisant les pondérations TF-IDF des documents comme entrées :

- Les arbres de décision en utilisant le critère *"Gini"* et sans imposer une profondeur maximale des arbres. Le nombre minimal utilisé pour les échantillons est de 2;
- Random Forest avec les mêmes paramètres utilisés dans la méthode précédente;
- le Bagging en utilisant les arbres de décision comme estimateur prinicipal (au nombre de 10);
- le GradientBoosting avec la régression logistique comme fonction d'erreur et une cadence d'apprentissage de 0,1;
- Support Vector machine (SVM) en utilisant RBF comme noyau et un paramètre de régularisation de 1,0.

Nous prenons aussi des approches neuronales comme éléments de comparaison pour l'évaluation de nos approches :

- **CNN** [Kim 2014] et **CNN-RNN** [Chen et al. 2017] qui utilisent des réseaux de neurones à convolutions pour extraire les caractéristiques propres au texte;
- **SGM** [Yang et al. 2018] qui applique un modèle de génération de séquences avec une nouvelle structure de décodeur pour résoudre le problème de CMLT;
- **MAGNET** [Pal et al. 2020] un réseau de graphes implémentant le mécanisme d'attention pour saisir la structure de dépendance entre les labels;
- **DocBERT** [Adhikari et al. 2019] : un fine-tuning des versions *base* et *large* de BERT pour la classification de documents.

5.3.2 L'optimal à atteindre. Les optimaux théoriques cibles (Approches oracles) se résument par les deux approches suivantes :

- L'approche oracle pour les N plus grandes activation, où nous considérons le nombre de labels présents pour une instance comme étant une donnée, pour ensuite prendre les N plus grandes activations comme labels présents;
- L'approche oracle pour les deux méthodes SCut de seuillage, où le calcul du seuil global et les seuils individuels est fait à partir du jeu de données de test. Cela pour avoir les résultats optimaux vers lesquels les deux méthodes de seuillage doivent s'approcher le plus possible.

5.4 Résultats

Dans cette partie, nous présentons les performances de toutes les approches citées dans les sections précédentes, testées sur les différents transformeurs présentés dans la section 5.1. Nous proposons les notations suivantes pour ces approches :

- "*SGO*" pour la méthode du Seuil Global Optimal (cf. section 3.2.1);00
- "*SI*" pour la méthode Seuil individuels (cf. section 3.2.2);
- "*NPA*" pour désigner l'approche des N plus grandes activations (cf. section 4.1);
- "*TL*" pour dénoter l'approche "couche de seuillage" (cf. section 4.2).

Les approches oracles quant à elles seront désignées en ajoutant *oracle* à la suite des approches concernées : SGO_{oracle}, SI_{oracle} et NPA_{oracle}.

La longueur maximale des séquences considérée est de 512 tokens ainsi qu'un *batch size* de 8, et ce pour tous les jeux de données. Les tableaux 2 et 4 présentent les scores de micro-F1 (avec précision et rappel) et Accuracy pour le jeu de données de test des corpus en anglais et français respectivement. Les tableaux 3 et 5 présentent les optimaux à atteindre des différentes approches pour les corpus en anglais et celui en français. Les résultats des méthodes de seuillage ainsi que les architectures proposées ont été obtenus à partir des versions *base* des transformeurs utilisés, qui diffère de la version *large* de part le nombre des couches transformeurs (12 pour

Table 2: Scores pour les données de test issues des corpus en anglais Reuters et AAPD (les meilleurs scores sont en bleu gras).

Modèles	Reuters				AAPD			
	Pr.	R	F1	Acc	Pr.	R	F1	Acc
Decision Tree	78,36	75,05	76,67	74,23	49,67	46,8	48,19	26,6
Bagging	88,12	79,65	83,67	73,34	77,27	48,16	59,34	26,9
Random Forest	**97,18**	57,13	71,96	64,06	**94,2**	25,49	40,12	20
GradientBoost	88,06	80,56	84,14	74,23	79,73	46,8	58,98	27,1
SVM	94,19	79,62	86,29	80,64	80,85	59,98	68,86	36,2
CNN	-	-	86,3	-	-	-	66,4	-
CNN-RNN	-	-	85,5	-	-	-	66,9	-
SGM	-	-	-	-	-	-	71,0	-
MAGNET	-	-	89,9	-	-	-	69,6	-
DocBERT$_{base}$	-	-	89,0	-	-	-	73,4	-
DocBERT$_{large}$	-	-	90,7	-	-	-	**75,2**	-
Méthodes de seuillage, sans architecture spécifique (SGO et SI)								
BERT+*SGO*	90,0	91,74	90,86	86,45	75,56	**72,65**	74,08	41,7
BERT+*SI*	88,89	**92,30**	90,56	85,72	75,51	72,61	74,04	41,3
DistilBERT+*SGO*	90,83	90,78	90,80	86,29	75,73	72,08	73,86	39,8
DistilBERT+*SI*	88,40	91,66	90,0	85,29	79,06	68,31	73,3	41,01
RoBERTa+*SGO*	90,73	89,90	90,31	86,28	74,49	72,49	73,47	40,3
RoBERTa+*SI*	89,77	90,49	90,13	85,92	75,35	71,09	73,15	40,2
DeBERTa+*SGO*	91,63	90,41	**91,02**	**86,78**	74,46	73,56	74,01	39,4
DeBERTa+*SI*	90,67	90,92	90,79	86,61	75,87	72,08	73,92	40,7
Adaptation architecture Transformeurs (NPA et TL)								
BERT+*NPA*	92,33	85,87	88,98	85,92	73,48	66,38	69,75	40,3
BERT+*TL*	90,60	90,41	90,50	86,12	73,48	72,20	72,83	39,5
DistilBERT+*NPA*	92,08	86,03	88,95	86,15	73,93	66,29	69,90	41,8
DistilBERT+*TL*	90,0	89,66	89,83	85,89	73,63	72,32	72,97	40,2
RoBERTa+*NPA*	89,43	83,20	86,20	83,40	75,04	66,58	70,56	42,0
RoBERTa+*TL*	91,17	89,05	90,09	86,02	76,48	71,58	73,94	**42,5**
DeBERTa+*NPA*	92,22	86,41	89,21	86,35	75,32	65,67	70,16	41,9
DeBERTa+*TL*	90,97	90,43	90,70	86,12	75,97	71,70	73,78	41,8

Table 3: Scores des approches oracles pour les données de test des corpus en anglais Reuters et AAPD (les meilleurs scores sont en bleu gras).

Modèles	Reuters				AAPD			
	Pr.	R	F1	Acc	Pr.	R	F1	Acc
BERT+*SGO*$_{oracle}$	91,46	90,70	91,08	87,01	80,41	68,85	74,18	43,1
BERT+*SI*$_{oracle}$	93,61	91,24	92,41	87,94	**83,0**	70,59	**76,29**	44,9
DistilBERT+*SGO*$_{oracle}$	91,61	90,17	90,88	86,52	74,50	73,27	73,88	40,0
DistilBERT+*SI*$_{oracle}$	92,48	92,04	92,26	87,45	82,59	70,34	75,97	44,0
RoBERTa+*SGO*$_{oracle}$	91,21	89,56	90,38	86,32	74,12	71,83	72,96	39,9
RoBERTa+*SI*$_{oracle}$	93,18	90,57	91,86	87,78	81,95	70,14	75,58	44,2
DeBERTa+*SGO*$_{oracle}$	92,74	90,73	91,72	87,41	75,21	71,95	73,55	40,4
DeBERTa+*SI*$_{oracle}$	**93,74**	91,56	**92,64**	88,27	82,82	70,26	76,02	44,0
BERT+*NPA*$_{oracle}$	92,55	**92,55**	92,55	**92,45**	74,06	74,06	74,06	51,8
DistilBERT+*NPA*$_{oracle}$	91,99	91,99	91,99	91,95	75,09	**75,09**	75,09	**54,7**
RoBERTa +*NPA*$_{oracle}$	91,69	91,69	91,69	91,45	73,36	73,36	73,36	51,5
DeBERTa +*NPA*$_{oracle}$	92,31	92,31	92,31	92,41	73,48	73,48	73,48	52,7

Table 4: Scores pour les données de test issues du corpus en français HAL-Dataset (les meilleurs scores sont en bleu gras).

Modèles	HAL-Dataset			
	Pr.	R	F1	Acc
Decision Tree	56,96	52,42	54,60	62,27
Bagging	86,24	52,66	65,39	62,35
Random Forest	90,72	50,85	65,17	64,60
GradientBoost	64,14	57,32	60,54	60,58
SVM	**95,23**	55,56	70,18	66,37
Seuillage sans architecture spécifique (SGO et SI)				
CamemBERT+*SGO*	79,17	67,11	72,64	71,84
CamemBERT+*SI*	77,12	67,65	72,08	71,35
FlauBERT+*SGO*	80,09	69,58	**74,47**	73,12
FlauBERT+*SI*	78,61	**70,01**	74,06	**73,45**
Adaptation architecture Transformeurs (NPA et TL)				
CamemBERT *NPA*	66,45	49,93	57,02	64,36
CamemBERT *TL*	88,38	59,31	70,98	68,78
FlauBERT *NPA*	65,70	60,58	63,04	67,42
FlauBERT *TL*	82,49	66,08	73,38	71,6

Table 5: Scores des approches oracles pour les données de test du corpus en français HAL-Dataset(les meilleurs scores sont en bleu gras).

Modèles	HAL-Dataset			
	Pr.	R	F1	Acc
CamemBERT+SGO_{oracle}	83,20	65,59	73,36	71,92
CamemBERT+SI_{oracle}	**90,33**	67,83	77,48	72,96
FlauBERT+SGO_{oracle}	83,16	68,38	75,05	73,61
FlauBERT+SI_{oracle}	89,21	70,49	**78,75**	75,30
CamemBERT+NPA_{oracle}	70,80	70,80	70,80	75,46
FlauBERT+NPA_{oracle}	72,85	**72,85**	72,85	**76,75**

la version base contre 24 pour la version large), ainsi que la taille du vecteur d'embedding (768 pour la version base contre 1024 pour la version large). L'entraînement des modèles a été réalisé à l'aide d'une Nvidia RTX 2080 Ti (11 Go de mémoire vidéo), le tableau 6 présente les différentes durées des phases d'entraînement et de validation pour les jeux de données étudiés.

Les modèles BERT et dérivés surpassent toutes les autres méthodes, que ce soient les méthodes classiques comme les SVM ou le GradientBoost, ou les autres méthodes d'apprentissage profond. Les méthodes de seuillage (*SGO* et *SI*) ainsi que les architectures proposées (*NPA* et *TL*) dépassent la version *base* de *DocBERT* (état de l'art actuel de la classification multilabels de textes pour les corpus AAPD et Reuters), et dans certains cas sa version *large*. L'implantation des méthodes *SGO* et *SI* sont les plus performantes parmi toutes les approches étudiées, avec un score micro-F1 de **91,02** et **90.79** respectivement, comparés au score de **90,7** de la version large de *DocBERT* pour le corpus *Reuters* (cf. tableau 2).

Table 6: Durée (en secondes) d'une itération dans la phase d'entraînement et de l'inférence pour la phase de validation des modèle BERT et FlauBERT sur les jeux de donnés AAPD, Reuters et HAL avec un batch size de 8

Modèles	AAPD		Reuters	
	Train	Valid	Train	Valid
BERT+*SGO*	2822	13,5	315	26,4
BERT+*SI*	3202	13,5	420	26,5
BERT+*NPA*	2031	13,7	223	27
BERT+*TL*	2015	25	213	50,8
	HAL			
	Train	Valid		
FlauBERT+*SGO*	539	15.6		
FlauBERT+*SI*	836	15.7		
FlauBERT+*NPA*	369	16.1		
FlauBERT+*TL*	379	29.6		

Ces deux méthodes améliorent de façon notable le score micro F1 et l'exactitude (*Accuracy*) des modèles, mais sont celles qui ont les coûts d'entraînement les plus élevés. Ceci est dû au fait que le calcul des seuils dans la phase d'entraînement requiert un temps de traitement considérable (cf. tableau 6). Le(s) seuil(s) de détection baisse parfois, ce qui a pour effet d'augmenter le nombre de vrais positifs; dans d'autre cas, ce(s) seuil(s) augmente, ce qui diminue le nombre de faux positifs. La précision du modèle augmente ainsi que la micro-F1 et l'exactitude.

Pour ce même corpus, l'architecture *TL* dépasse la version *base* du modèle *DocBERT* ainsi que le modèle *MAGNET*, et se rapproche de la version *large* du premier avec un score de **90,60** pour le modèle *DeBERTa*. L'approche *NPA* quant à elle ne réussit pas à atteindre son optimal théorique, en obtenant un score quand bien même dépassant la version *base* de *DocBERT* (cf. partie Adaptation architecture Transformeur du tableau 2). Ces deux architectures nécessitent un temps d'entraînement moins important que les deux méthodes de seuillage (une différence de 1000 secondes dans la configuration de nos tests pour le modèle BERT) , mais possèdent un coût d'inférence plus élevé, notamment pour l'architecture *TL* où le temps d'inférence, **25** et **29.6** secondes pour BERT et FlauBERT respectivement, est presque deux fois celui des autres méthodes (cf. tableau 6). Cela peut être expliqué par la présence de la couche supplémentaire que possède cet architecture, augmentant ainsi le temps d'inférence des modèles, chose qui n'impacte pas de façon considérable la phase d'entraînement de par sa nature parallèle (l'inférence étant exécutée d'une manière séquentielle couche après couche).

Pour le corpus AAPD, les méthodes de seuillage ainsi que l'approche *TL* réussissent à obtenir de bonnes performances comparées aux autres méthodes, mais le modèle $DocBERT_{large}$ reste le plus performant parmi toutes les approches étudiées. La nature complexe du vocabulaire scientifique de ce corpus souligne l'apport que peut avoir l'augmentation de la taille des modèles transformeurs (rajout de plusieurs couches et augmentation de la taille des embeddings).

L'approche SI_{oracle} est la plus performante des approches oracles étudiées. En effet, le calcul d'un seuil individuel pour chaque label à partir du jeu de test conduit vers un optimal général pour toutes les

classes qui dépassent l'optimal de la méthode *SGO*. Mais cela n'est pas le cas pour son équivalent expérimental. *SGO* reste la méthode qui s'approche le plus de son optimal théorique (cf. tableaux 3 et 2). Chose qui peut être expliquée par le fait que pour la méthode *SI* plusieurs seuils doivent être calculés pour chaque label à partir du jeu d'entraînement, ce qui ne garantit par un optimal général sur l'ensemble des classes pour le jeu de test. Cet effet est accentué si le nombre de classes est grand.

Pour l'optimal théorique de la méthode *NPA*, fournir le nombre effectif de labels pertinents pour une instance permet de considérer présents de nouveaux labels, ce qui augmente le taux de vrais positifs et baisse celui des faux négatifs. Mais cela a pour effet d'augmenter aussi le risque de faux positifs car ces nouveaux labels sont dans certains cas des prédictions non valides. Les instances pour lesquelles cette approche réduit le nombre de labels précédemment prédits sont rares. Donc le score micro F1 reste inchangé ou peut baisser par rapport aux optimaux théoriques des approches de seuillage. D'autre part, une augmentation considérable de l'exactitude est constatée, c'est d'ailleurs l'approche qui réussit à obtenir les scores d'exactitude les plus élevés (cf. tableau 3 et 5). L'implantation effective de cette approche quant à elle n'a pas permis d'obtenir les même performances que son optimal théorique. L'état caché de la phrase contenu dans le token [CLS] n'est peut-être pas suffisant pour le calcul du nombre de classes présentes. Dans le futur, l'exploitation des scores d'attention obtenus dans les différentes couches du modèle pourrait être une méthode plus efficace pour accomplir cette tâche.

On note aussi que les deux architectures proposées obtiennent des scores de micro-précision plus élevés que les approches de seuillage. Pour l'approche *NPA*, cela est dû au fait que la sélection des labels est réalisée suivant l'approximation du nombre de classes calculé. Pour *TL* cela peut être dû au fait que les valeurs des activations des labels considérés comme présents sont accentuées, contre l'atténuation des labels considérés comme non pertinents.

Les mêmes tendances sont observées pour notre corpus de texte en français *"HAL-Dataset"*. Les méthodes de seuillage obtiennent les scores micro-F1 les plus élevés, **74,47** pour *SGO* et **74,06** pour *SI*, (cf. tableau 4) suivies de l'architecture *TL* avec un score de **73,38**. Un score proche des méthodes précédentes avec une architecture qui ne requiert pas une phase de recherche de seuils optimaux.

On note tout de même que l'approche *NPA*, que ce soit l'optimal théorique ou l'implantation expérimentale, n'est pas aussi performantes que les autres méthodes. Cela pourrait être dû à la difficulté de l'apprentissage du nombre de label compte tenu du déséquilibre du jeu de données. Environ 82% des instances n'ont qu'un seul label (cf. figure 4).

DeBERTa est la variante de *BERT* la plus performante parmi tous les modèles évalués, les changements qu'apportent cette variante compte tenu de l'ajout de l'information concernant la position relative du mot par rapport à la phrase, semblent améliorer les performances de *BERT*. Cette amélioration du rendement vient au détriment de la vitesse d'entraînement et d'inférence du modèle qui devient plus lent que les autres variantes. *DistilBERT* quant à lui obtient des résultats au même niveau que sa version originale, malgré sa taille réduite (moins de couches transformeurs). La distillation des connaissances semble être une méthode efficace pour pallier à l'inconvénient majeur des transformeurs et des réseaux neuronaux : la nécessité d'un temps d'entraînement très élevé.

Pour la tâche de classification multilabels de texte, *RoBERTa* n'est pas aussi performant que les autres variantes. Cela peut être dû à l'absence de la partie "prédiction de la phrase suivante", une partie qui peut avoir une importance pour la classification multilabels, dans la mesure où elle peut permettre au modèle d'apprendre la manière dans les idées d'un même domaine se succède et de pouvoir faire la différence entre plusieurs champs lexicaux.

CamemBERT qui est une variante basée sur *RoBERTa*, hérite des mêmes problématiques face à la classification multilabels, et se voit dépassée par *FlauBERT* dont la force réside dans le fait qu'il soit entraîné sur des corpus très variés, avec des styles d'écritures diverses.

On note aussi que les SVM et les forêts d'arbres décisionnels obtiennent les scores de micro-précision les plus élevés, au détriment du taux de rappel, faisant ainsi baisser le score micro F1. Mais la précision n'est pas un facteur suffisant pour la mesure des performances. SVM peut être considéré comme l'approche non neuronale la plus performante en vu de son score d'exactitude élevé, mais qui reste en dessous des autre méthodes testées.

6 CONCLUSION ET TRAVAUX FUTURS

Les modèles de langue à base de transformeurs surpassent les autres architectures neuronales profondes et constituent une base solide adaptable pour une multitude de tâches de traitement automatique des langues et la classification de textes, c'est la direction que nous avons suivie dans cette étude.

Tout d'abord, nous avons testé et montré dans cet article que les approches de seuillage sont très performantes pour la classification multilabels. Le calcul d'un seuil global obtient des résultats plus élevés que le calcul d'un seuil individuel pour chaque classe. L'optimisation faite sur chaque classe ne garantit pas un optimal général pour tous les labels. Nous avons par la suite proposé des modifications sur l'architecture des transformeurs. Ceci par le rajout d'une couche supplémentaire, avec un nombre d'activations égal au nombre de classes, à l'aval du modèle pour s'affranchir d'une optimisation sur les seuils. Une approche qui, en moyenne, est aussi performante que les approches de seuillage. L'étude des optimaux à atteindre a montré que l'utilisation du nombre de classes pour la sélection des labels pertinents augmente de façon considérable les performances de la classification multilabels. Cependant, la différence entre ces optimaux et les résultats effectifs de nos expérimentations montrent que notre proposition doit être améliorée dans le cas des jeux de données déséquilibrés. Au final les deux familles d'approches, seuillage et modification de l'architecture des transformeurs, sont des méthodes qui améliorent les performances des réseaux d'apprentissage profond pour la classification multilabels de texte.

La langue des corpus de texte, anglais pour AAPD et Reuters, ou français pour le corpus HAL-Dataset que nous avons construit, ainsi que leur nature (article scientifiques ou dépêches journalistiques) ne semblent pas être des facteurs qui impactent les performances des approches proposées. Ce sont d'ailleurs des approches qui peuvent être utilisées pour tout problème de classification multilabels. Dans le contexte applicatif de nos travaux de thèse, nous construisons un

jeu de données à partir des mails clients de l'agence, et sur lequel nous testerons toutes ces approches.

Chaque variante de *BERT* cherche à améliorer les aspects contraignants de ce modèle. *DistilBERT*, malgré sa taille réduite, obtient des résultats aussi élevés que la version originale. *DeBERTa* se positionne comme étant la variante la plus performante, en compagnie de *FlauBERT* qui dépasse *CamemBERT*, la version basée sur *RoBERTa* dont les performances restent derrière les autres variantes.

Nous avons montré l'intérêt réel que peuvent avoir les approches d'exploitation des activations des couches de sorties. D'autres méthodes d'adaptation des réseaux neuronaux pour la classification multilabels peuvent être explorées, dans la mesure où les seuils de détection peuvent être des paramètres appris du modèle lors de la phase d'entraînement. Des améliorations peuvent être faites sur les architectures alternatives au seuillage proposées, notamment la recherche d'une méthode plus efficace pour le calcul du nombre de classes présentes pour la méthode *NPA* ainsi que l'augmentation du nombre de couches de seuillage pour l'approche *TL*. Toutes ces approches peuvent être utilisées pour l'optimisation de tout modèle d'apprentissage profond.

En ce qui concerne le domaine d'application, la classification multilabels est une tâche pertinente dans le monde industriel. Cependant la CMLT n'est pas une tâche si fréquente et on regrette qu'elle ne figure pas parmi les comparatifs les plus en vue comme GLUE.

REFERENCES

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. DocBERT: BERT for Document Classification. *arXiv:1904.08398 [cs]* (April 2019). http://arxiv.org/abs/1904.08398 arXiv: 1904.08398 version: 1.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]* (May 2016). http://arxiv.org/abs/1409.0473 arXiv: 1409.0473.

Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. 2004. *Learning multi-label scene classification.*

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and I. Dhillon. 2019. X-BERT: eXtreme Multi-label Text Classification with using Bidirectional Encoder Representations from Transformers. *undefined* (2019). https://www.semanticscholar.org/paper/X-BERT%3A-eXtreme-Multi-label-Text-Classification-Chang-Yu/3b9efab2114a3456dccb9d625ca732100d18ba74

Guibin Chen, Deheng Ye, Zhenchang Xing, Jieshan Chen, and Erik Cambria. 2017. *Ensemble application of convolutional and recurrent neural networks for multi-label text categorization.* https://doi.org/10.1109/IJCNN.2017.7966144 Pages: 2383.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]* (May 2019). http://arxiv.org/abs/1810.04805 arXiv: 1810.04805.

Luyun Gan, Brosnan Yuen, and Tao Lu. 2019. Multi-label Classification with Optimal Thresholding for Multi-composition Spectroscopic Analysis. *arXiv:1906.10242 [cs, eess, stat]* (June 2019). http://arxiv.org/abs/1906.10242 arXiv: 1906.10242.

Jibing Gong, Zhiyong Teng, Qi Teng, Hekai Zhang, Linfeng Du, Shuai Chen, Md Zakirul Alam Bhuiyan, Jianhua Li, Mingsheng Liu, and Hongyuan Ma. 2020. Hierarchical Graph Transformer-Based Deep Learning Model for Large-Scale Multi-Label Text Classification. *IEEE Access* 8 (2020), 30885–30896. https://doi.org/10.1109/ACCESS.2020.2972751 Conference Name: IEEE Access.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *arXiv:2006.03654 [cs]* (Jan. 2021). http://arxiv.org/abs/2006.03654 arXiv: 2006.03654.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural computation* 9 (Dec. 1997), 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. *arXiv:1408.5882 [cs]* (Sept. 2014). http://arxiv.org/abs/1408.5882 arXiv: 1408.5882.

S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics* 22, 1 (March 1951), 79–86. https://doi.org/10.1214/aoms/1177729694 Publisher: Institute of Mathematical Statistics.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15).* AAAI Press, Austin, Texas, 2267–2273.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2020. FlauBERT: Unsupervised Language Model Pre-training for French. *arXiv:1912.05372 [cs]* (March 2020). http://arxiv.org/abs/1912.05372 arXiv: 1912.05372.

David Lewis, Ctr Info, Lang Studies, and Marc Ringuette. 1996. A Comparison of Two Learning Algorithms for Text Categorization. *Third Annual Symposium on Document Analysis and Information Retrieval* (Oct. 1996).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]* (July 2019). http://arxiv.org/abs/1907.11692 arXiv: 1907.11692 version: 1.

Oscar Luaces, Jorge Díez, José Barranquero, Juan José del Coz, and Antonio Bahamonde. 2012. Binary relevance efficacy for multilabel classification. *Progress in Artificial Intelligence* 1, 4 (Dec. 2012), 303–313. https://doi.org/10.1007/s13748-012-0030-x Number: 4.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a Tasty French Language Model. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020), 7203–7219. https://doi.org/10.18653/v1/2020.acl-main.645 arXiv: 1911.03894.

Min-Ling Zhang and Zhi-Hua Zhou. 2006. Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization. *IEEE Transactions on Knowledge and Data Engineering* 18, 10 (Oct. 2006), 1338–1351. https://doi.org/10.1109/TKDE.2006.162 Number: 10 Conference Name: IEEE Transactions on Knowledge and Data Engineering.

Ankit Pal, Muru Selvakumar, and Malaikannan Sankarasubbu. 2020. Multi-Label Text Classification using Attention-based Graph Neural Network. *Proceedings of the 12th International Conference on Agents and Artificial Intelligence* (2020), 494–505. https://doi.org/10.5220/0008940304940505 arXiv: 2003.11644.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv:1802.05365 [cs]* (March 2018). http://arxiv.org/abs/1802.05365 arXiv: 1802.05365.

Alec Radford, Jeff Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019). /paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d6169988371b2755e573cc28650d14dfe

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108 [cs]* (Feb. 2020). http://arxiv.org/abs/1910.01108 arXiv: 1910.01108.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. *arXiv:1409.3215 [cs]* (Dec. 2014). http://arxiv.org/abs/1409.3215 arXiv: 1409.3215.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. Leibniz-Institut für Deutsche Sprache. https://doi.org/10.14618/IDS-PUB-9021

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2010. Mining Multi-label Data. In *Data Mining and Knowledge Discovery Handbook*, Oded Maimon and Lior Rokach (Eds.). Springer US, Boston, MA, 667–685. https://doi.org/10.1007/978-0-387-09823-4_34

Grigorios Tsoumakas and Ioannis Vlahavas. 2007. Random k-Labelsets: An Ensemble Method for Multilabel Classification. In *Machine Learning: ECML 2007 (Lecture Notes in Computer Science)*, Joost N. Kok, Jacek Koronacki, Raomon Lopez de Mantaras, Stan Matwin, Dunja Mladenič, and Andrzej Skowron (Eds.). Springer, Berlin, Heidelberg, 406–417. https://doi.org/10.1007/978-3-540-74958-5_38

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *arXiv:1706.03762 [cs]* (Dec. 2017). http://arxiv.org/abs/1706.03762 arXiv: 1706.03762.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. SGM: Sequence Generation Model for Multi-label Classification. *arXiv:1806.04822 [cs]* (June 2018). http://arxiv.org/abs/1806.04822 arXiv: 1806.04822.

Yiming Yang. 1997. *An evaluation of statistical approach to text categorization.* Technical Report.

Yiming Yang. 2001. A study of thresholding strategies for text categorization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '01).* Association for Computing Machinery, New York, NY, USA, 137–145. https://doi.org/10.1145/383952.383975

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* Association for Computational Linguistics, San Diego, California, 1480–1489. https://doi.org/10.18653/v1/N16-1174

Min-Ling Zhang and Zhi-Hua Zhou. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition* 40, 7 (July 2007), 2038–2048. https://doi.org/10.1016/j.patcog.2006.12.019 Number: 7.

IOPE: Interactive Ontology Population and Enrichment Guided by Ontological Constraints

Shadi Baghernezhad-Tabasi[1], Loïc Druette[2], Fabrice Jouanot[1],
Celine Meurger[2], Marie-Christine Rousset[1,3]
[1]Université Grenoble Alpes, CNRS, LIG, Grenoble, France, [2]Université Claude Bernard Lyon 1, SAMSEI, Lyon, France,
[3]Institut Universitaire de France, Paris, France
[1]firstname.lastname@univ-grenoble-alpes.fr, [2]firstname.lastname@univ-lyon1.fr

ABSTRACT

In this paper, we focus on the construction of specialized ontologies that capture skills of experienced experts in a particular domain with the goal of sharing them with a larger community of trainees or less experienced experts in the domain. Our main contribution is the automatic construction of a Graphical User Interface (GUI) named IOPE built from the ontological constraints of an input ontology, as the support for the controlled update process of the considered ontology. The resulting GUI functions as a guidance for the experts with no knowledge of OWL/RDFS, which enables them to easily explore and update their ontologies. We illustrate the functionality of IOPE on an ontology for simulation-based medical workshops called OntoSAMSEI.

1 INTRODUCTION

Ontologies are the backbone of many information systems that require access to structured knowledge. By their very nature, real-world ontologies are dynamic artifacts that evolve both in their structure (the data model) and their content (instances). Keeping them up-to-date is a critical operation for most applications which rely on semantic Web technologies. Ontology updates encompass both *enrichment* and *population*. Ontology updates are often performed manually, as the non-documented knowledge of the domain expert is required to be taken into consideration. However, this manual updates put burden on the experts and render the whole ontological ecosystem inefficient. In this paper, we advocate for an alternative and more effective approach, and propose to handle updates automatically through a few interactions with the expert, using a Graphical User Interface (GUI).

The challenges associated to interaction-based automatic updates are two-fold: (*i*) While ontologies are typically represented in the form of graphs, it is inherently difficult and counter-intuitive to provide a graphical graph-based representation of ontologies for the consumption of experts. While there exist several methods to visualize a graph structure [1, 2], the outcome is often hard to digest by domain experts. (*ii*) It is unclear how experts should perform ontology updates through the interactions, without the prior knowledge of the formal syntax and the semantics of ontology languages.

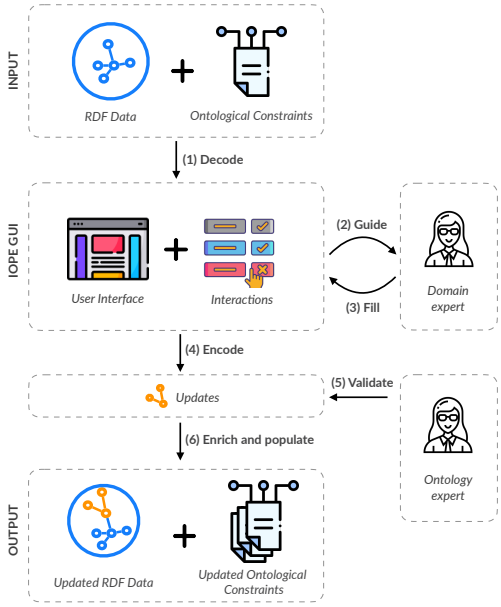


Figure 1: Overview of IOPE workflow

In this paper, we demonstrate IOPE (Interactive Ontology Population and Enrichment), a framework for the automatic construction of a GUI using *prefilled Web forms*. We leverage Web forms as a natural interaction means to tackle the challenge of counter-intuitive ontology representations. IOPE generates the Web forms from *ontological constraints*, which support the controlled update process of a given ontology, and prefills the generated forms. While IOPE is generic and can be applied to ontologies from a variety of domains, we employ an ontology called OntoSAMSEI [3] for demonstration purposes, whose content helps the domain experts design teaching units for learning skills in simulation-based Medicine. OntoSAMSEI's IOPE GUI is accessible via the following link: http://iope.tabasi.info (in French).

2 INTERACTIVE ONTOLOGY UPDATE

Our approach consists of transposing the RDF data and the ontological constraints of a given domain ontology into a GUI named IOPE GUI. It functions as a guidance for domain experts to easily explore the ontology and update it through interactive graphical widgets. The input entered by domain experts through the IOPE

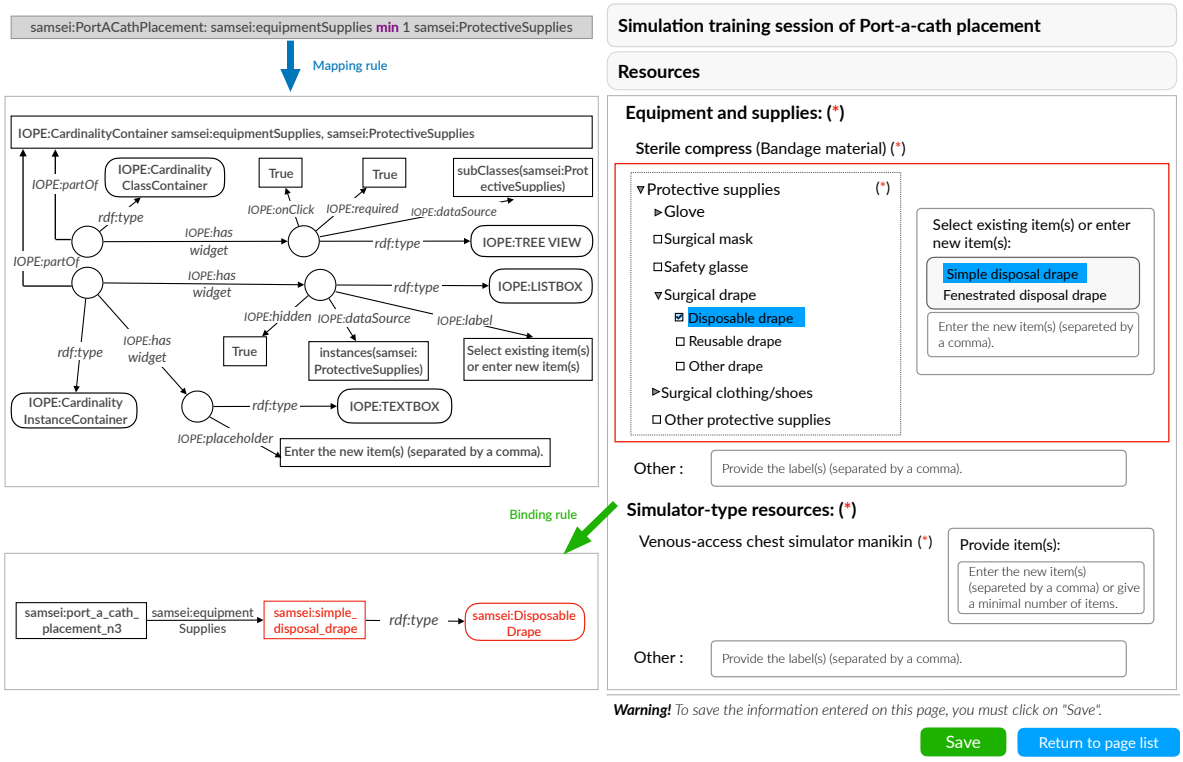


Figure 2: An instance of a pre-filled Web page generated by applying the mapping rules to ontological constraints, and the transformation of interactions with the binding rules.

GUI is transformed into RDF triples that must be verified by an ontology engineer before being added effectively in the domain ontology. Figure 1 is an overview of our interactive IOPE system. In the following, we describe the components of IOPE workflow.

Input: A domain ontology is the input for GUI construction. The ontological constraints of the input ontology have been automatically saturated by a reasoning algorithm, detailed in [4].

IOPE GUI: The GUI consists of automatically generated pre-filled Web pages which represent domain ontological constraints. We specify a set of 16 *mapping rules* to generate pre-filled Web pages [4]. The domain expert is guided using the generated pages, where she interacts with the GUI to explore and update the ontology through interactive graphical widgets. Those interactions must be bound to RDF data corresponding to new instances or new constraints submitted for populating or enriching the domain ontology. We specify a set of 9 *binding rules* to generate RDF graphs from user entries [4]. The resulting RDF graphs should be verified by an ontology engineer before adding to the domain ontology.

Output: The output is the updated RDF data and the updated ontological constraints.

Figure 2-right illustrates one of the resulting pre-filled Web pages generated by applying the mapping rules to OntoSAMSEI's ontological constraints, to represent information about the workshop required resources, e.g., equipment and supplies, and simulator-type resources. At the top-left of the figure, an ontological constraint is mentioned in Turtle format[1], conveying that at least one equipment of type "protective supply" is mandatory for this workshop. Obviously, the Turtle format is difficult to be comprehended by users without OWL/RDFS knowledge. The graph-based result of a mapping rule applied on the constraint is shown in the middle-left part of the figure, whose HTML implementation is highlighted with a red-border square on the right side. The user interactions with the pre-filled Web page are highlighted in blue, where the educator selects a sub-class Disposable Drape from the Protective Supplies hierarchy, and picks the instance Simple disposable drape from the provided list of instances. The interactions are transformed into RDF graphs using the binding rule shown in the bottom-left part of the figure. This transformation results in two RDF triples which declare relations between an instance of the workshop and the selected instance, and between the selected instance and its class.

REFERENCES

[1] Yixiang Fang, Reynold Cheng, Siqiang Luo, Jiafeng Hu, and Kai Huang. C-Explorer: Browsing communities in large graphs. *VLDB*, 2017.
[2] Nathalie Henry, Jean-Daniel Fekete, and Michael J. McGuffin. Nodetrix: a hybrid visualization of social networks. *TVCG*, 2007.
[3] Shadi Baghernezhad-Tabasi, Loic Druette, Fabrice Jouanot, Celine Meurger, and Marie-Christine Rousset. OntoSAMSEI: Interactive ontology engineering for supporting simulation-based training in medicine. In *WETICE*, 2021.
[4] Shadi Baghernezhad-Tabasi, Loïc Druette, Fabrice Jouanot, Celine Meurger, and Marie-Christine Rousset. IOPE: Interactive ontology population and enrichment guided by ontological constraints. In *WISE*, 2021.

[1] *https://www.w3.org/TR/turtle/*

5 Résumés des articles courts

Digital Preservation with Synthetic DNA

Eugenio Marinelli
eugenio.Marinelli@eurecom.fr
EURECOM
France

Eddy Ghabach
eddy.ghabach@eurecom.fr
EURECOM
France

Thomas Bolbroe
tbo@sa.dk
Rigsarkivet
Denmark

Omer Sella
osella@imperial.ac.uk
Imperial College
London, UK

Thomas Heinis
t.heinis@imperial.ac.uk
Imperial College
London, UK

Raja Appuswamy
raja.appuswamy@eurecom.fr
EURECOM
France

ABSTRACT

The growing adoption of AI and data analytics in various sectors has resulted in digital preservation emerging as a cross-sectoral problem that affects everyone from data-driven enterprises to memory institutions alike. As all contemporary storage media suffer from fundamental density and durability limitations, researchers have started investigating new media that can offer high-density, long-term preservation of digital data. In the European Union-funded Future and Emerging Technologies project OligoArchive, we are exploring one such media, namely, synthetic Deoxyribo Nucleic Acid (DNA). In this paper, we provide an overview of the ongoing collaboration between project OligoArchive and the Danish National Archive in preserving culturally important digital data with synthetic DNA.

1 INTRODUCTION

Today, we live in an increasingly digital society. Thus, preservation of digital data has emerged as an important problem. In order to preserve digital data, it is necessary to first store the data safely over a long time frame. Historically, this task has been complicated due to several issues associated with digital storage media. All current media technologies suffer from density scaling limitations resulting in storage capacity improving at a much slower rate than the rate of data growth. All current media also suffer from media decay that can cause data loss due to silent data corruption, and have very limited lifetime compared to the requirements of digital preservation. In project OligoArchive, we are exploring a radically new storage media that has received a lot of attention recently—Deoxyribo Nucleic Acid (DNA). DNA possesses several key advantages over current storage media.It is an extremely dense three-dimensional storage medium, very durable and can last millennia in a cold, dry, dark environment. In this work, we provide an overview of the ongoing collaboration between the Danish National Archive and project OligoArchive in demonstrating a holistic solution for long-term preservation of culturally significant data using DNA.

2 DESIGN

The Danish National Archives is a knowledge center documenting the historical development of the Danish society. A huge part of their work includes the preservation of digitally created and retro-digitized data securely and cost-effectively. As the vast majority of data in the Danish public sector are organized as databases with or without files in various formats, the focus has been on archiving these data in a standardized, system-independent and cost efficient manner. As a result, the archive has implemented a Danish version of the SIARD format named SIARD-DK for storing of such data. SIARD is an open format, designed for archiving relational database snapshots in a vendor-neutral form and is used in the CEF building block “eArchiving”.

The archival material used for this work consists of selected hand-drawings made by the Danish king Christian IV and its related database information in SIARD format. The first step in preserving data is taking a snapshot of the database and creating an SIARD-DK Archival information Package (AIP). In the creation of this particular AIP, the digitized image was converted to TIFF format. Information relevant to the images stored in a database such as the preservation format of the files, their title, creator and original size, descriptive information, etc., was extracted and packaged together with relevant documentation in the AIP-format. The resulting AIP is a single ZIP64 file that internally contains the TIFF images, in addition to XML and XSD files that store the schema of the archive and metadata information. Traditionally, the SIARD file is stored on tape, which is internally organized as a sequence of blocks. Thus, the main research challenges are (i) developing a reliable block abstraction for DNA, (ii) developing a serialization and coding strategy for mapping the binary SIARD file to DNA blocks.

2.1 DNA data storage pipeline

The end-to-end DNA media storage pipeline is presented in Figure 1. In the rest of this section, we will provide an overview of both the write path that takes as input the SIARD zip file and stores it in DNA, and the read path that restores back the zip file from DNA.

2.1.1 Write Path. In order to store the archive on synthetic DNA, the zip file is first encoded from binary into a quaternary sequence of oligonucleotides, and then synthesized to generate synthetic DNA. During encoding, the file is read as a stream of bits and pseudo randomized. Randomization is applied to reduce the number of homopolymer repeats within oligos, and similarity across oligos when generating the oligos at a later stage. After randomization,

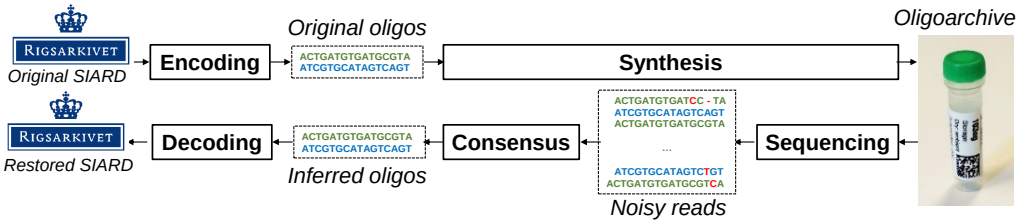


Figure 1: DNA Storage Pipeline

error correction encoding is applied to protect the data against errors. We use large-block length Low-Density Parity Check (LDPC) codes with a block size of 256,000 bits as the error correction code, as it has been shown to be able to recover data in the presence of intra-oligo errors, or even if entire oligos are missing. We configure LDPC to add 10% redundancy to convert each sequence of 256,000 bits into 281,600 bits with data and parity. Each 281,600 bit sequence is then used to generate a set of 300-bit sequences, where each 300-bits is composed of 281 data bits and a 19-bit index that is used to order the sequences. Each 300-bit sequence is then passed to a constrained code that converts it into an oligonucleotide sequence.

We provide a practical overview of the constrained code here, deferring rigorous mathematical definition to future work. The constrained code is essentially a finite state machine that views each oligo as a sequence of short symbols that are concatenated together. In our current configuration, the constrained code breaks up each 300-bit sequence into a series of ten 30-bit integers. Each 30-bit integer is fed as input to a deterministic finite state machine that takes as input the previous 16-nucleotide sequence, and a set of constraints (homopolymer repeat, GC ratio, etecetera) that each symbol must meet and produces a valid 16-nucleotide sequence corresponding to the 30-bits as output. Thus, each 300-bit sequence is encoded as concatenation of ten such symbols, each with a length of 16 nucleotides, leading to an oligo that is 160 nucleotides long. We would like to explicitly point out here that the length of an oligo is a configurable parameter.

2.1.2 Read Path. To retrieve back the SIARD archive, the DNA is sequenced in order to retrieve back the nucleotide sequence of oligos. As mentioned before, sequencing produces noisy copies of the original oligos that can contain insertion, deletion, or substitution errors, which are referred to as *reads*. In order to infer the original oligos from the reads, we use a consensus procedure. Concretely, we structure this process as sequence of three algorithms. First, we identify all pairs of strings that are similar to each other. As modern sequencers produce hundreds of millions of reads, this first task is an extremely computationally intensive due to use of edit distance as a metric for comparing strings. Thus, we have developed an efficient similarity join algorithm, called OneJoin, that exploits the fact that due to randomization during encoding, reads corresponding to the same original oligo are "close" to each other despite some errors and "far" from the reads related to other oligos. The results obtained from the join algorithm are then used to quickly identify clusters of strings that are similar to each other. Each cluster thus groups all reads belonging the same oligo. Finally, we apply a position-wise consensus procedure that uses multiple reads to infer the original oligo in each cluster using a sequence alignment procedure. We would like to point out here that not all oligos need to be correctly inferred. In fact, some original oligos might not appear at all in the inferred set, and other inferred oligos might have errors. We rely on the parity added by LDPC codes at a higher level to recover data despite these errors.

The inferred oligos are then passed to the decoder which reverses the encoding steps. The constrained code is first used to convert each 160nt oligo back into 300 bit sequences. The index stored in each 300 bits is used to reassemble bits back in the correct order. The LDPC decoder is then used to recover back data even if some bits were wrongly decoded, or some bits were zeroed out as corresponding oligos were missing. The decoded data is then derandomized to obtain a stream of bits that corresponds to the SIARD zip file.

3 EVALUATION

We provide a preliminary, simulation driven evaluation where we encode/decode the real dataset using our pipeline. The raw SIARD archive that is fed as input to our pipeline is 12.9MB in size. With redundancy added by LDPC, the resulting binary data to be stored on DNA is 14.19MB in size. We encode the SIARD archive generating 404,863 oligos, each with a length of 160 nucleotides. We then generate five million reads by using a short-read simulator tool, that adds random errors such as insertion, substitution, and deletion in each read to mimic the actions of an Illumina DNA sequencer. This corresponds to an average coverage of 11×, meaning that each oligo, on average is covered by 11 noisy copies. Using the consensus procedure described earlier, we obtain the inferred oligos using this simulated dataset. We then use the constrained code to convert these inferred oligos into 300-bit sequences, and reassemble them in order based on the 19-bit index. At this stage, we can have a situation where an oligo is missing but we rely on the redundancy added by LDPC to recover back the original archive.

4 CONCLUSION

In this work, we provided an overview of the ongoing collaboration between project OligoArchive and the Danish National Archive in using DNA to preserve culturally significant digital data. Building on prior work on molecular information storage and digital preservation, we presented a holistic, end-to-end pipeline for preserving both data and the meaning of data on DNA, and tested the pipeline using simulation studies.

Vers une modélisation du paysage médiatique français

Définir, observer et modéliser les médias d'information

Agnès Saulnier
Institut National de l'Audiovisuel
Bry Sur Marne, France
asaulnier@ina.fr

RESUME

Internet a bousculé l'univers médiatique et pour s'y retrouver l'enjeu est de fournir un nouvel ensemble de critères de description des entités médiatiques. D'où vient l'information ? Comment est-elle financée ? Comment a-t-elle été produite ? Dans quel but ? Par qui est-elle diffusée ? Pour quel public ? Autant de questions essentielles à se poser pour mieux se diriger dans l'écosystème des médias. Notre objet d'étude porte ainsi sur la définition, l'observation et la modélisation des médias d'informations. Ce papier se concentre plus particulièrement sur la démarche de modélisation reposant sur une grille d'observation des entités médiatiques. Modéliser le domaine médiatique est en effet une tâche complexe et délicate car elle doit prendre en compte un niveau très fin de description des médias ainsi que l'aspect diachronique lié à l'évolution de la structure juridique des entités médias, de la structure de production et de la ligne éditoriale. Le modèle est testé et affiné à l'aide d'une liste de 3000 sites web d'information.

1 Introduction

L'enjeu de ce travail est de modéliser conjointement les médias de masse traditionnels et les nouveaux médias très divers que permet Internet, afin d'aider les utilisateurs à s'orienter dans le nouveau paysage médiatique. En effet, de nos jours, Internet véhicule un grand nombre d'informations qu'il est important de pouvoir distinguer car le terme information tend à se confondre avec celui de contenu, faisant disparaitre la valeur ajoutée journalistique. Notre démarche repose sur la définition du périmètre, la constitution d'une grille d'observation et la modélisation conceptuelle du domaine médiatique. Cette modélisation nécessite une plus grande finesse de description que celle des systèmes déjà existants ainsi que la prise en compte de l'aspect diachronique lié à l'évolution des médias. Une liste de 3000 sites web d'information a été recensée afin d'aider à la normalisation du vocabulaire et aux tests du modèle. Cette modélisation est destinée à déboucher plus tard sur la réalisation d'une base de connaissance médiatique.

2 Contexte

Avant de modéliser un domaine, il convient de définir son périmètre [1]. Celui-ci ne se limite pas aux médias traditionnels (presse écrite, télévision, radio) mais inclut aussi tous les nouveaux médias liés à Internet. Benoit Lafon a utilisé le concept de médiatisation pour élargir le périmètre «La médiatisation consiste en la mise en média d'individus, de groupes ou d'institutions par la construction de produits médiatiques formalisés, dans une visée stratégique, impliquant des pratiques collectives de consommation» [2, p163]. De nombreux chercheurs en sciences de l'information et de la communication ont qualifié les dimensions constitutives des médias. Force est de constater que le terme média renvoie généralement à cinq référents distincts (technique, contenu, organisation, modèle économique, usage) sur lesquels peut reposer la grille d'observation. En ce qui concerne les ontologies déjà existantes, Emilio Sanfilippo [3] a effectué un état de l'art assez général de celles dédiées aux entités d'information. Plusieurs sont spécialisées dans les systèmes documentaires (Dublin Core, CIDOC CRM, FRBR, $PRESS_{00}$, BibFrame…) mais aucune ne répond totalement à nos besoins de finesse de description et de diachronicité.

3 Grille d'observation

La grille d'observation détaille cinq aspects des médias. L'aspect technique décrit les supports, formats et moyens de diffusion. L'aspect contenu repose de son coté sur la ligne éditoriale du média. C'est elle qui définit le fond des contenus proposés, qui est lié au sens, ainsi que leur forme, qui est liée à la présentation. La presse dispose déjà de nombreux systèmes de classification, suivant la cible du public, la thématique, la périodicité et le périmètre. Il existe aussi un grand nombre de familles de presse qui proposent des classifications exclusives [4], tout comme pour la radio et la télévision. En revanche, pour Internet, il n'existe pas de

famille officiellement reconnue. Nous pouvons nous inspirer de la littérature [5] pour proposer une taxonomie. L'aspect organisation recouvre le cadre légal dans lequel le média doit exercer son activité. Il est important de distinguer les éditeurs de contenu des éditeurs de services. L'aspect économique décrit quant à lui les modèles de financement des médias (lecteurs, annonceurs, subventions), le modèle d'économie de l'attention ainsi que les solutions adoptées par les médias face au bouleversement médiatique (optimisation des coûts, concentration des médias, diversification des supports et services…). Enfin l'aspect usage repose sur la finalité des messages transmis par les médias (intérêt social, politique et financier) qui peut être discernée à partir des genres de contenu et des types de producteurs d'information.

4 Modélisation du domaine médiatique

Le modèle doit décrire de manière homogène des médias très divers et tenir compte de contraintes spécifiques :

- Niveau très fin de description (blog d'un site de presse).
- Aspect diachronique (fusion de médias).
- Notion de provenance (plusieurs familles de presse).

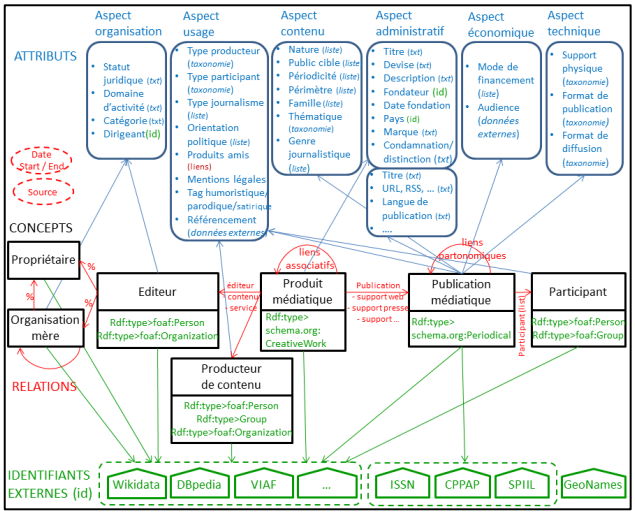


Figure 1 : Schéma général du modèle conceptuel

Le modèle proposé est représenté sur la Figure 1. Les concepts principaux (en noir), que l'on retrouve aussi sous le nom de classes, correspondent tout d'abord aux éléments de la chaine de production médiatique: Producteur de contenu, Editeur, Participant, Maison mère et Propriétaire. Ensuite, nous nous sommes inspirés des ontologies documentaires, BibFrame et FRBR, pour distinguer le Produit médiatique (œuvre) de ses Publications, ce qui permet de relier plusieurs publications (presse papier, web, ou réseaux sociaux…) à un même produit. Les identifiants externes (en vert) font référence à des éléments de base du Web sémantique. Il s'agit aussi bien de classe pour le modèle que d'URIs pour identifier des ressources particulières dans les instances. Il existe plusieurs sortes de relations (en rouge) : des relations « partonomiques » (*tout-partie*) qui permettent de décrire finement une publication ; des relations associatives qui permettent d'associer plusieurs concepts entre eux ; des relations hiérarchiques *(« est un »)* qui peuvent décrire des liens d'appartenance pour définir une typologie. Les relations peuvent aussi être spécialisées en sous-relations. Pour terminer, les attributs (en bleu) s'appuient sur les différents aspects des médias que nous avons vus dans la grille. En ce qui concerne la modélisation du temps, il s'agit de poser un intervalle temporel sur chaque propriété.

Une liste d'autorité de 3000 sites web d'information a été constituée en croisant des sources de données de différents types (CPPAP, BNF, SPIIL, Upreg, ACPM, annuaires…). Cette liste d'autorité, destinée à fournir des instances dans la future base de connaissance, a servi à deux types de test. Tout d'abord, une sélection d'une vingtaine de sites représentatifs de la liste a permis d'évaluer la structure générale et la souplesse du modèle conceptuel. Ensuite, 2/3 de la liste a été analysé à la main pour tester, affiner et compléter les attributs. Ainsi des critères de la presse écrite ont pu être étendus à d'autres supports. Des typologies ont pu être affinées à partir de l'observation de la liste. Les tests ont aussi démontré le besoin d'ajouter des descripteurs particuliers.

5 Conclusion

Les journalistes professionnels n'ont plus le monopole de l'information, l'éditeur de contenu se voit concurrencé par des éditeurs de service, et les supports de diffusion se sont multipliés. Afin de pouvoir modéliser ce nouveau paysage médiatique protéiforme, il faut tout d'abord définir son périmètre, puis observer son fonctionnement, identifier les entités à décrire. La particularité du modèle obtenu repose sur un très fin niveau de description et sur l'évolution diachronique des entités. La démarche mise en place s'appuie sur trois spécificités : l'analyse très fine du domaine médiatique, l'utilisation de ressources disponibles sur le web (relations, classes, URIs) et le développement à la main de typologie à l'aide d'une liste d'autorité. Dans un futur travail, ce modèle devra être implémenté sous la forme d'une base de données ou d'une base de connaissances. La recherche diachronique sera une question importante à évaluer.

REFERENCES

[1] Agnès Saulnier, 2021, Eléments de description du paysage médiatique français, HAL-03343071
[2] Benoit Lafon, 2019, Médias et médiatisation. Analyser les médias imprimés, audiovisuels, numériques, Communication en +, Presses universitaires de Grenoble, 310p
[3] Emilio Sanfilippo, 2021, Ontologies for information entities: State of the art and open challenges, Applied Ontology, vol. 16, no. 2, pp. 111-135. DOI: 10.3233/AO-210246
[4] Patrick Eveno, 2010, La presse, Paris: PUF, p 45
[5] Emmanuel Marty, Franck Rebillard, Stéphanie Pouchot, Thierry Lafouge, 2012, Diversité et concentration de l'information sur le web, Une analyse à grande échelle des sites d'actualité français, Réseaux 2012/6, N°176, pp 27-72.

Practical Fully-Decentralized Secure Aggregation for Personal Data Management Systems

Julien Mirval
julien.mirval@cozycloud.cc
Cozy Cloud
Inria-Saclay
UVSQ, Université Paris-Saclay
France

Luc Bouganim
luc.bouganim@inria.fr
Inria-Saclay
UVSQ, Université Paris-Saclay
France

Iulian Sandu-Popa
iulian.sandu-popa@uvsq.fr
UVSQ, Université Paris-Saclay
Inria-Saclay
France

Personal Data Management Systems (PDMS) are flourishing, boosted by legal and technical means like smart disclosure, data portability and data altruism. A PDMS allows its owner to easily collect, store and manage data, directly generated by her devices, or resulting from her interactions with companies or administrations. PDMSs unlock innovative usages by crossing multiple data sources from one or many users, thus requiring aggregation primitives. Indeed, aggregation primitives are essential to compute statistics on user data, but are also a fundamental building block for machine learning algorithms. This paper proposes a protocol allowing for secure aggregation in a massively distributed PDMS environment, which adapts to selective participation and PDMSs characteristics, and is reliable with respect to failures, with no compromise on accuracy. Preliminary experiments show the effectiveness of our protocol which can adapt to several contexts with varying PDMSs characteristics in terms of communication speed or CPU resources and can adjust the aggregation strategy to the estimated selective participation.

The new privacy-protection regulations (e.g., GDPR) and smart disclosure initiatives in the last decade have boosted the development and adoption of Personal Data Management Systems (PDMSs) [1]. A PDMS (e.g., Cozy Cloud, Nextcloud, Solid) is a data platform allowing users to easily collect, store and manage into a single place data directly generated by user devices (e.g., quantified-self data, smart home data, photos) and data resulting from user interactions (e.g., social interaction data, health, bank, telecom). Users can then leverage the power of their PDMS to benefit from their personal data for their own good and in the interest of the community [2].

Consequently, the PDMS paradigm leads to an important shift in the personal data ecosystem since data becomes massively distributed, at the user-side. It also holds the promise of unlocking innovative usages. An individual can now cross her data from different data silos, e.g., health records and physical activity data. Moreover, individuals can cross data within large communities of users, e.g., to compute statistics for epidemiological studies or to train a machine learning model (ML) for recommender systems or automatic classification of user data. However, these exciting perspectives should not eclipse the security issues –user data must be kept private– and the right for any PDMS user to consent, or not, in participating in each computation.

Aggregation primitives (e.g., sum or average) are obviously essential to compute basic statistics on user data but are also a fundamental building block for machine learning algorithms. Thus, to enable such new usages, we need scalable, privacy-preserving protocols implementing data aggregation primitives with selective (i.e., consenting) participants. Ideally, the proposed protocol should provide an accurate result that fully takes advantage of high-quality data available in PDMSs. Efficiency (i.e., protocol latency and total load of the system) is of prime importance and the protocol should adapt to several contexts: the PDMSs could be limited by their communication speed or by their computation power. Finally, given the scale of such decentralized aggregation, such protocols must also be robust to node failures. To summarize, our goal is to propose an aggregation protocol for basic aggregate functions that fulfills the following properties:

- *fully decentralized and highly scalable*, with the number of participants.
- *privacy-preserving*, i.e., it protects the confidentiality of user data.
- *accurate*, i.e., it does not require a trade-off between accuracy and privacy.
- *adaptable*, i.e, it can adapt to a large spectrum of computation selectivity values (reflecting the subset of contributor nodes) and system configurations (network and cryptographic latency).
- *reliable*, i.e., it handles node failures or voluntary disconnections.

REFERENCES

[1] Nicolas Anciaux, Philippe Bonnet, Luc Bouganim, Benjamin Nguyen, Philippe Pucheral, Iulian Sandu Popa, and Guillaume Scerri. 2019. Personal data management systems: The security and functionality standpoint. *Information Systems* 80 (2019), 13–35.

[2] EU Commission. 25 October 2020. Proposal for a Regulation on European data governance (Data Governance Act), COM/2020/767. [eur-lex].

Un cadre orienté graphe bien fondé pour le résumé de bases de données en logiques de description

Cheikh-Brahim El Vaigh
Univ. Rennes
Lannion, France
cheikh-brahim.el-vaigh@irisa.fr

François Goasdoué
Univ. Rennes
Lannion, France
fg@irisa.fr

ABSTRACT

L'opération de quotient de la théorie des graphes offre un cadre élégant pour le résumé de grands graphes; cette opération consiste à fusionner des noeuds équivalents au sens d'une relation d'équivalence. Le résumé de graphes fondé sur cette opération a été largement étudié dans la littérature, notamment pour l'exploration et la gestion efficace de grands graphes.

Nous étudions si une opération similaire peut être utilisée pour résumer des bases de données en logiques de description (LD), c'est-à-dire des ABoxes. Dans ce but, nous définissons et examinons l'opération de quotient sur une ABox: nous établissons qu'une ABox quotient est plus spécifique que la ABox qu'elle résume et nous caractérisons dans quelle mesure elle est plus spécifique. Ces investigations préliminaires valident l'intérêt d'un cadre de résumé de ABoxes fondé sur l'opération de quotient de graphe, et ouvrent la voie à de futurs travaux en LD, par exemple pour concevoir des relations d'équivalence adaptées à l'optimisation des tâches typiques de gestion et de raisonnement sur de grandes ABoxes ou encore à la visualisation de grandes ABoxes, ainsi qu'à l'application de ces travaux dans des domaines connexes comme le Web Sémantique.

Toward Generic Abstractions for Data of Any Model

Nelly Barret
Inria & Institut Polytechnique de Paris
nelly.barret@inria.fr

Ioana Manolescu
Inria & Institut Polytechnique de Paris
ioana.manolescu@inria.fr

Prajna Upadhyay
Inria & Institut Polytechnique de Paris
prajna-devi.upadhyay@inria.fr

ABSTRACT

Digital data sharing leads to unprecedented opportunities to develop data-driven systems for supporting economic activities, the social and political life, and science. Many open-access datasets are RDF graphs, but others are CSV files, Neo4J property graphs, JSON or XML documents, etc.

Potential users need to *understand* a dataset in order to decide if it is useful for their goal. While some datasets come with a schema and/or documentation, this is not always the case. Data summarization or schema inference tools have been proposed, specializing in XML, or JSON, or the RDF data models. In this work, we present a *dataset abstraction approach*, which (*i*) applies on relational, CSV, XML, JSON, RDF or Property Graph data; (*ii*) computes an *abstraction meant for humans* (as opposed to a *schema meant for a parser*); (*iii*) integrates Information Extraction data profiling, to also *classify* dataset content among a set of categories of interest to the user. Our abstractions are conceptually close to an Entity-Relationship diagram, if one allows nested and possibly heterogeneous structure within entities.

Efficiently identifying disguised nulls in heterogeneous text data

Théo Bouganim
Inria & IPP
France
theo.bouganim@inria.fr

Helena Galhardas
INESC-ID & IST, Univ. Lisboa
Portugal
hig@inesc-id.pt

Ioana Manolescu
Inria & IPP
France
ioana.manolescu@inria.fr

ABSTRACT

Digital data is produced in many data models, ranging from highly structured (typically relational) to semi-structured models (XML, JSON) to various graph formats (RDF, property graphs) or text. Most real-world datasets contain a certain amount of *null* values, denoting missing, unknown or unapplicable information. While some data models allow representing nulls by special tokens, so-called *disguised nulls* are also frequently encountered: these are values that are not syntactically speaking nulls, but which do, nevertheless, denote the absence, unavailability or unapplicability of the information.

This paper describes our ongoing work toward detecting disguised nulls in textual data, encountered in ConnectionLens graphs. Driven by journalistic applications, we focus for now on large, semistructured datasets, where most or all data values are free-form text. We show that the state-of-the-art methods for detecting nulls in relational databases, mostly tailored towards numerical data, do not detect disguised nulls efficiently on such data. Then, we present two alternative methods: (i) leveraging Information Extraction, and (ii) text embeddings and classification. We detail their performance-precision trade-offs on real-world datasets.

The paper is available at https://hal.inria.fr/hal-03347947v1

ACKNOWLEDGMENTS

This work was supported by the ANR AI Chair project SourcesSay Grant no ANR-20-CHIA-0015-01.

Using projection to improve Differential Privacy on RDF graphs

Sara Taki
INSA Centre Val de Loire, Laboratoire d'Informatique Fondamentale d'Orléans
Bourges, France
sara.taki@insa-cvl.fr

Cedric Eichler
INSA Centre Val de Loire, Laboratoire d'Informatique Fondamentale d'Orléans
Bourges, France
cedric.eichler@insa-cvl.fr

Benjamin Nguyen
INSA Centre Val de Loire, Laboratoire d'Informatique Fondamentale d'Orléans
Bourges, France
benjamin.nguyen@insa-cvl.fr

ABSTRACT

Differential privacy is one of the most popular and prevalent definitions of privacy, providing a robust and mathematically rigid definition of privacy. In the last decade, adaptation of differential privacy to graph data has received growing attention. Most efforts have been dedicated to unlabeled graphs, homogeneous graphs, while labeled graphs with an underlying semantic have been mildly addressed.

In this paper, we present a new approach based on graph projection to adapt differential privacy to edge-labeled directed graphs, i.e. RDF graphs, while reducing query sensitivity. We propose three edge-addition based graph projection methods that transform the original RDF graph into a graph with bounded degree, bounded out-degree, and bounded typed-out-degree and characterize their influence on differential privacy. We thus propose a general method to expand the domain of any differentially private algorithm from graphs with bounded (out/typed-out) degree to any arbitrary RDF graph. We illustrate our approach under a realistic tweeter use-case.

1 ACKNOWLEDGMENT

Work supported by the French National Research Agency, under grant ANR-18-CE23-0010

Towards a Logistical View for Data Lake Optimization

Marzieh Derakhshannia*
Anne Laurent*
Marzieh.derakhshannia@etu.umontpellier.fr
Anne.laurent@umontpellier.fr
LIRMM, Univ Montpellier, CNRS
Montpellier, France

ABSTRACT

Data life cycle management is a challenging issue in the big data arena. Data lakes are the new generation data repositories that provide the platform to respond to the management of huge amounts of raw data. In terms of structure, data lake architecture could have a significant role in improving the data quality and service levels. For this reason, suitable data lake design and appropriate management strategies play the important role in implementing of fruitful data lake. Logistical view is a proper solution to manage an optimal data lake that will be favorable for organisations.

CCS CONCEPTS

• **Information systems** → **Storage architectures**; **Data management systems**.

KEYWORDS

Data Lake, Data Lake Architecture, Logistic Systems, Supply Chain, Supply Chain Management

1 CONTEXT

In the last decades, the data has become a valuable property for organizations and is considered as digital oil in the big data environment. To make capital from this precious product, proper equipment is required [6]. Traditional data management tools have inadequacies in gathering, storing, processing, and visualizing all types of data in an integrated manner in the big data arena. Therefore, the term " Data lake " has been emerged to address the shortcomings flaws and meet all the challenges related to the data lifecycle [10].

Data lakes are the developed, affordable, and agile generation of centralized data management systems that provide plenty of faculties to conduct the procedures of storage and knowledge mining from a huge amount of heterogeneous data [4, 9, 12]. However, welcoming all kinds of data in their native formats increases the risk of saturation with unmanageable or useless data; a phenomenon is known as “data swamp”; and threatens the security of sensitive data. For this reason, data management strategies and architecture design are more highlighted in data lake productivity. The architecture of the data lake and the administration regulations could have significant impacts on the efficiency of data lake performance. Thus, an optimized architecture under proper managerial disciplines enhances the data lake credibility and reinforces its reliability as a new component of the information systems to support decision-making processes in the organizations.

2 CHALLENGES

Data lake architecture design and data management are problematic issues for the implementation of an efficient data lake infrastructure. Based on these requirements, many studies in the field of the data lake architecture design have been conducted in parallel with the technical implementation [5, 7–9, 11, 12]. Despite many types of researches on the data lake architecture and with regard to their cons and pros, optimization of the data lake structure and performance based on managerial and mathematical viewpoints has been less considered. For this reason, we suggest logistical approaches like product lifecycle management and hybrid decision-making strategies (like facility location-allocation problems) based on supply chain management principles to design and optimize the data lake architecture. The significant principle of supply chain management is to minimize the total costs related to the flow of products in such a way that profitability of the chain and quality of the service requirement, will be maximized. We intend to implement this principle for data lake management due to the new proposed architecture and imitated strategies.

3 METHODS

Since the data could be considered as a valuable product, we can imagine that a data lake performs like a logistic system that manages the data throughout its life cycle. This viewpoint brings about a great deal of systematic and managerial strategies to develop data lake architecture and gain the integrated manner to manage and govern the data in a cost-effective platform. Based on this requirement, the supply chain is one of the well-known logistic systems that present a systematic view of logistic parties who are integrated into a unified network to produce a commodity or provide a service [1, 2]. To design the profitable supply chain network, each level of this chain is managed by tactical and strategic strategies. Thus, the supply chain management strategies prepare the suitable guidelines to design and manage the data lake architecture in general or in detail scales.

The processes of designing a data lake architecture are related to positioning the different components like data ingestion, data storage, data process, and data access stages and defining the proper strategies and protocols to govern the data like metadata management or data democracy. For this reason, the data lake platform based on systematic definitions and network design concepts could

*Both authors contributed equally to this research.

be an effective step in the evolution of data lake architecture. Regarding the analogy of data lake and supply chain, we consider the data like a product or service that should be monitored through several stages in the data lake architecture from data entering until data consumption. With respect to this point of view, we can benefit from some management tools to improve data lake functionality. For example, we can use the policies of product lifecycle assessment that is employed in logistic systems, for data governance in data lake or hybrid decision-making problems that are essential in supply chain network design, for implementation and optimization of the data lake platform.

As figure 1 shows the simple analogy of supply chain structure and data lake architecture, data are supplied, stored, manipulated, monitored, and delivered in data lake as a product that is managed throughout the supply chain. In this architecture, data are produced and supplied from different sources and are entered in the data lake by the ingestion stage, then they are stored in their raw formats at the storage level, and finally, they are sent to the processing stage for further procedures based on users requirements. In this architecture, each level of data life cycle could be managed by mimetic logistical patterns. For example, the ingestion strategies are related to the optimal decisions about metadata management and data modeling, like supply and product design management in the supply chain. The storage strategies concentrate on selecting cost-effective storage strategies (single-store systems or multi-store systems) and data integration, just like decision-making about the number, the location, and the type of the warehouses and inventory management in the supply chain. The process strategies are the vital decisions that impact on quality of data and veracity of extracted knowledge. Therefore, the main goal of process strategies is related to choose the data processing methods that prepare the data with the fastest time and the least cost possible for data queries. Based on this requirement, the hybrid decision-making models like location-allocation problems; that bring about the best solution to optimize the supply chain networks [3]; could be a logical solution to design the affordable data lake architecture. For instance, we can use the techniques of supply chain network design (like optimization mathematical models) to determine the structure of data lake with a variety of strategic decisions such as determining the number of proper processing jobs, tactical decisions like data governance all over the data lake just like product quality assessment in the supply chain, and operational decisions like fulfilling user demands. The main objective of designing the logistical data lake architecture is to minimize further costs related to the data ingestion, storage, and processing and maximize the level of data lake performance.

4 CONCLUSION

In the direction of mentioned goals, we focus on data lake architecture design due to the mathematical optimization models that are inspired by hybrid problems in supply chain network design. The accomplished studies showed that these interdisciplinary methods smooth the path of presenting data lake optimization models and guarantee to obtain the affordable data lake architecture as a logistic system.

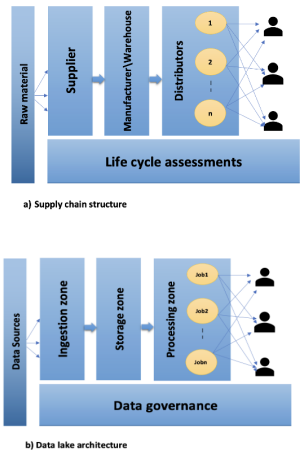


Figure 1: Analogy of supply chain structure and data lake architecture

REFERENCES

[1] Benita M Beamon. 1998. Supply chain design and analysis:: Models and methods. *International Journal of Production Economics* 55, 3 (1998), 281 – 294. https://doi.org/10.1016/S0925-5273(98)00079-6

[2] Sunil Chopra and Peter Meindl. 2007. Supply chain management. Strategy, planning & operation. In *Das summa summarum des management*. Springer, Germany, Springer Gabler, 265–275.

[3] Mark S Daskin. 2011. *Network and discrete location: models, algorithms, and applications*. John Wiley & Sons, 111 River St, 07030 Hoboken.

[4] Huang Ling Fang. 2015. Managing data lakes in big data era: What's a data lake and why has it became popular in data management ecosystem. *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)* „ 15503297 (2015), 820–824.

[5] Corinna Giebler, Christoph Gröger, Eva Hoos, Rebecca Eichler, Holger Schwarz, and Bernhard Mitschang. 2021. The Data Lake Architecture Framework: A Foundation for Building a Comprehensive Data Lake Architecture. In . Conference for Database Systems for Business, Technology and Web (BTW), 70469 Stuttgart, Germany. https://doi.org/10.18420/btw2021-19

[6] Dennis D Hirsch. 2013. The glass house effect: Big Data, the new oil, and the power of analogy. *Me. L. Rev.* 66 (2013), 373.

[7] Bill Inmon. 2016. *Data Lake Architecture: Designing the Data Lake and avoiding the garbage dump*. Technics publications, .

[8] Jabrane Kachaoui and Abdessamad Belangour. 2020. From single architectural design to a reference conceptual meta-model:an intelligent data lake for new data insights. *International Journal* 8, 4 (2020), 1460–1465.

[9] Alice LaPlante and Ben Sharma. 2016. *Architecting data lakes: data management architectures for advanced business use cases*. O'Reilly Media, 1005 Gravenstein Highway North, 95472 Sebastopol.

[10] Pasupuleti Pradeep. 2015. *Data lake development with big data : explore architectural approaches to building Data Lakes that ingest, index, manage, and analyze massive amounts of data using Big Data technologies*. Packt Publishing, Birmingham.

[11] Pegdwendé Sawadogo and Jérôme Darmont. 2021. On data lake architectures and metadata management. *Journal of Intelligent Information Systems* 56, 1 (2021), 97–120.

[12] John Tomcy. 2017. *Data Lake for enterprises*. Packt Publishing, Birmingham.

6 Résumés des articles de démonstration

Un Outil de Génération de Témoins pour les schémas JSON

A Tool for JSON Schema Witness Generation

Lyes Attouche
Université Paris-Dauphine, PSL Research University
lyes.attouche@dauphine.fr

Mohamed-Amine Baazizi
Sorbonne Université, LIP6 UMR 7606
baazizi@ia.lip6.fr

Dario Colazzo
Université Paris-Dauphine, PSL Research University
dario.colazzo@dauphine.fr

Francesco Falleni
Dipartimento di Informatica, Università di Pisa
fallenifrancesco98@gmail.com

Giorgio Ghelli
Dipartimento di Informatica, Università di Pisa
ghelli@di.unipi.it

Cristiano Landi
Dipartimento di Informatica, Università di Pisa
c.landi7@studenti.unipi.it

Carlo Sartiani
DIMIE, Università della Basilicata
carlo.sartiani@unibas.it

Stefanie Scherzinger
Universität Passau
stefanie.scherzinger@uni-passau.de

ABSTRACT

JSON Schema is an evolving standard for the description of JSON documents. It is an extremely powerful language endowed with boolean operators and recursive definitions. Hence, classical problems like schema *consistency* and *equivalence* may be challenging without well-principled tools. Based on our recent effort for laying down an algebraic formal semantics of JSON Schema, we demonstrate an approach for generating valid *witnesses* of a user-defined schema. Our goal is not only to allow programmers to design schemas that meet their intentions, but also to guide them in their journey to understanding the semantics of existing schemas, in an interactive fashion. We thus aim to contribute to the adoption of the JSON Schema language by facilitating its use.

1 INTRODUCTION

In recent years, JSON has become the *de facto* standard data interchange format, and is now widely used for exchanging data between web applications and remote servers, for exporting and importing data, as well as inside complex ML pipelines for combining different stages, as in Google TFX [8].

Despite its great popularity, there is no consensus about a *standard* schema language for JSON yet. Indeed, in many cases, JSON datasets come without a schema, and the end user or application has the duty to infer or guess a new schema, if required. In many other cases, however, several and vastly different schema languages are used for describing the structure of JSON data, ranging from Apache Avro [3], to the MongoDB internal schema language [5], and to JSON Schema [9].

Differently from what happened with XML, whose standard schema languages (DTDs and XML Schema) reached quickly a wide diffusion, JSON Schema is not being adopted at the same pace. Many reasons are slowing down its adoption, but, according to our observations, a major obstacle is the fact that, while extremely powerful, JSON Schema is – frankly – hard to use. Indeed, a schema is a logical combination of implicative assertions, and some of them may produce side effects on previous ones.

As a consequence, leaving the realm of plain vanilla schemas may expose the programmer to many risks, such as the definition of a schema with unintended semantics, or one that is even empty.

Example 1.1. Consider the following schema.

```
{
    "type": "object",
    "properties": {
        "x": { "type": "integer" }
    },
    "required": [ "x" ]
}
```

This schema declares that all instances are JSON objects, and that each object has a mandatory *member* whose name is x and whose type is *integer*. This schema, however, does not impose further constraints on object values; therefore, an object may also have supplementary and unconstrained members.

Example 1.2. Consider now the following schema, differing only in the next-to-last line.

```
{
    "type": "object",
    "properties": {
        "x": { "type": "integer" }
    },
    "not": { "required": [ "x" ] }
}
```

One may assume that this specifies that x is "not required", hence is optional. However, given the semantics of JSON Schema, negating a required member does not make it optional: indeed, the final effect is to actually forbid the presence of the member, hence excluding any JSON object having a member whose name is x (this example is inspired by a discussion on Stack Overflow [1], where the confusing effect of this schema is testified).

Given the complex and non-trivial interplay between schema assertions, designing a rich yet sound schema is challenging, especially when other powerful mechanisms of JSON Schema are involved, such as negation, mutual exclusion, recursion, union and conjunction, as well as array constraints controlling array length and content, possibly requiring uniqueness of elements.

Motivating Witness Generation. The state-of-the-art approach for exploring JSON Schema semantics is ultimately a manual trial and error: using a JSON Schema validator, a schema designer can

test whether a JSON document is valid w.r.t. the schema. That is, the designer must come up with suitable *witness* documents.

Yet, in this demo, we present a tool capable of automatic witness generation. For instance, for the schema from Example 1.1, our tool generates the witness `{"x": 0}`, as any valid instance must be an object, where member x is mandatory and integer-typed. For Example 1.2, our tool generates the witness `{}`, since the empty object is valid. For the schema designer, this valuable feedback may well increase the overall productivity.

Moreover, upon the push of a button, the designer can generate further witnesses. In the example just discussed, the designer would be provided with `{"0": null}`. Thus, by interactive iteration, the designer ensures that he or she "gets it right".

Moreover, our tool allows the comparison of two schemas: for a witness that is valid w.r.t. the schema from Example 1.2, but not w.r.t. the schema from Example 1.1, the tool returns the JSON document `{}`. For the other way round, a witness is `{"x": 0}`. Again, the designer can request further witnesses, as needed.

Contributions. The goal of this demonstration is to showcase a tool allowing the schema designer to investigate the formal properties of a schema, and even to compare schemas. Our tool is based on our earlier contributions on algebraic manipulations of JSON Schema [6]. With our tool, the designer can:

- obtain an algebraic representation of the input schema;
- generate a witness for the schema, to verify whether the schema is empty or not, and to gain insights into the actual semantics of a given schema;
- exploit witness generation for checking whether a schema S_1 is a subtype of a schema S_2, and hence, whether it represents a conservative and not disruptive evolution.

2 DEMONSTRATION OVERVIEW

Our demo can be tried online https://jsonschematool.ew.r.appspot.com/. Its setup includes two datasets: schemas from the JSON Schema Test Suite [2], a collection of small schemas that serve as unit tests for JSON Schema validators (and explore different operators), and real-world schemas from SchemaStore.org [4]. Naturally, our attendees may also formulate their own schemas.

Typically, the user will first enter a JSON Schema document (or load one of the provided schemas), and then convert the schema (shown in the midsection of our screenshot) into our algebra (shown in the bottom section). Our algebra has been designed to be close to the original language, to be intuitive for practitioners. Yet different from the JSON Schema language, our algebra enjoys substitutability, that is, the semantics of an operator does not depend on its context, which eases manipulation.

The user may then choose to generate a first JSON witness. If the system finds no witness, it will alert the user that the schema is empty, otherwise, a witness is generated.

If there is a witness, the user can generate a further ("yet another") witness, that is different from all those previously seen. Alternatively, the user can edit the original JSON Schema, or directly the algebraic expression, and request that a new first witness is generated (disregarding witnesses already seen).

The schema designer can choose to convert back from the algebra to JSON Schema. Thus, the schema designer can interactively explore the semantics of a given schema, switch between the JSON Schema representation and the (often more compact) representation in our algebra, and iteratively revise the schema.

To allow interested demo attendees to inspect the internals of witness generation, as outlined in the previous section, our tool

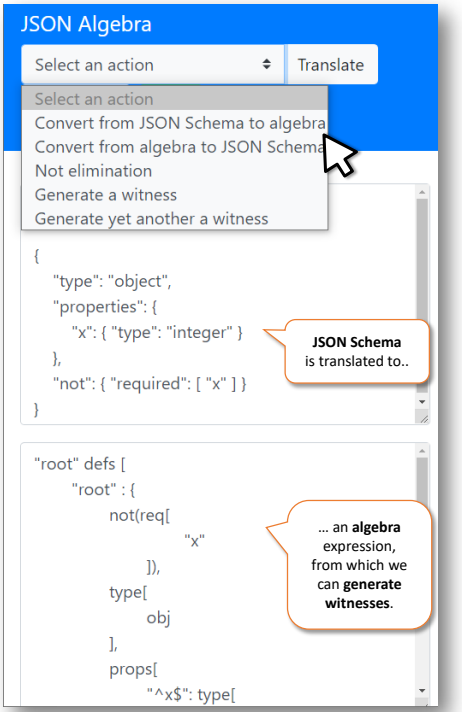


Figure 1: Screenshot: from JSON Schema to our algebra.

can also perform negation elimination on algebraic expressions. This feature would not be included in a tool targeted at end users.

Comparing schemas. Our tool offers a second screen (not shown here) where two schemas may be compared. Rather than computing a boolean answer to the question whether one schema subsumes the other, as done in state-of-the-art tools today [7], our tool can generate a witness that exemplifies a JSON document which is valid w.r.t. the one schema, but not the other.

ACKNOWLEDGMENTS

Giorgio Ghelli's contribution has been funded by MIUR project PRIN 2017FTXR7S "IT-MaTTerS" (Methods and Tools for Trustworthy Smart Systems). Stefanie Scherzinger's contribution has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 385808805.

REFERENCES

[1] [n.d.]. JSON Schema – valid if object does *not* contain a particular property. Available at:https://stackoverflow.com/questions/30515253/json-schema-valid-if-object-does-not-contain-a-particular-property.
[2] [n.d.]. JSON Schema Test Suite. Available at:https://github.com/json-schema-org/JSON-Schema-Test-Suite.
[3] 2020. Apache Avro. http://avro.apache.org.
[4] 2020. JSON Schema Schema. https://www.schemastore.org/json/.
[5] 2020. MongoDB. https://www.mongodb.com.
[6] Mohamed-Amine Baazizi, Dario Colazzo, Giorgio Ghelli, Carlo Sartiani, and Stefanie Scherzinger. 2020. Not Elimination and Witness Generation for JSON Schema. In *Proc. BDA 2020.*
[7] Michael Fruth, Mohamed-Amine Baazizi, Dario Colazzo, Giorgio Ghelli, Carlo Sartiani, and Stefanie Scherzinger. 2020. Challenges in Checking JSON Schema Containment over Evolving Real-World Schemas. In *Proc. EmpER.*
[8] Ion Stoica. 2020. Systems and ML: When the Sum is Greater than Its Parts. In *Proc. SIGMOD.*
[9] A. Wright, H. Andrews, and B. Hutton. 2019. *JSON Schema Validation: A Vocabulary for Structural Validation of JSON - draft-handrews-json-schema-validation-02.* Technical Report. Internet Engineering Task Force. https://tools.ietf.org/html/draft-handrews-json-schema-validation-02

Discovering Conflicts of Interest across Heterogeneous Data Sources with ConnectionLens

Angelos-Christos Anadiotis
École Polytechnique, IPP & EPFL
angelos.anadiotis@polytechnique.edu

Oana Balalau
Inria & IPP
oana.balalau@inria.fr

Théo Bouganim
Inria & IPP
theo.bouganim@inria.fr

Francesco Chimienti
Inria & IPP
francesco.chimienti@inria.fr

Helena Galhardas
INESC-ID & IST, Univ. Lisboa
hig@inesc-id.pt

Mhd-Yamen Haddad
Inria & IPP
mhd-yamen.haddad@inria.fr

Stéphane Horel
Le Monde
horel@lemonde.fr

Ioana Manolescu
Inria & IPP
ioana.manolescu@inria.fr

Youssr Youssef
Inria & IPP
youssr.youssef@gmail.com

ABSTRACT

Investigative Journalism (IJ, in short) requires **combining highly heterogeneous digital datasets** coming from a wide variety of sources. We have developed ConnectionLens, a system that integrates such sources into a single heterogeneous graph and enables users to query the graph using keywords. The first iteration of the system [3] followed a mediator architecture which severely constrained its query scalability. Thus, we **fully re-engineered the system**, moving it to a warehouse architecture, and replacing its core components (information extraction, data querying, and interactive interfaces), which allowed us to handle uses cases orders of magnitude larger than the previous platform [2]. In a consortium of computer scientists and investigative journalists, we propose to demonstrate ConnectionLens' capability to integrate arbitrary heterogeneous datasets and query them flexibly by means of keywords. Among several scenarios, our main focus will be on a **real-world journalistic use case** about situations which may lead to Conflicts of Interest between biomedical experts and various organizations, such as corporations, lobbies, etc. The demonstration will showcase the end-to-end data analysis pipeline, illustrate each system component, and the different parameters governing graph creation and querying.

The demonstration appears in CIKM 2021 [1]. A demonstration video is available at: https://youtu.be/5B0KRow0dv8.

Acknowledgments. The authors thank M. Ferrer and the Décodeurs team (Le Monde) for introducing us, and for many insightful discussions. We also thank Jérémie Feitz and Emmanuel Pietriga for their work on the GUI of ConnectionLens. This work was supported by the ANR AI Chair project SourcesSay Grant no ANR-20-CHIA-0015-01.

REFERENCES

[1] Angelos-Christos Anadiotis, Oana Balalau, Francesco Chimienti, Helena Galhardas, Mhd-Yamen Haddad, Stéphane Horel, Ioana Manolescu, and Youssr Youssef. 2021. Discovering Conflicts of Interest across Heterogeneous Data Sources with ConnectionLens. In *ACM International Conference on Information and Knowledge Managemetn (CIKM)*.

[2] Angelos Christos Anadiotis, Oana Balalau, Catarina Conceicao, Helena Galhardas, Mhd Yamen Haddad, Ioana Manolescu, Tayeb Merabti, and Jingmao You. 2021. Graph integration of structured, semistructured and unstructured data for data journalism. *Information Systems* (July 2021), 42. https://doi.org/10.1016/j.is.2021.101846

[3] Camille Chanial, Rédouane Dziri, Helena Galhardas, Julien Leblay, Minh-Huong Le Nguyen, and Ioana Manolescu. 2018. ConnectionLens: Finding Connections Across Heterogeneous Data Sources. *Proc. VLDB Endow.* 11, 12 (2018), 2030–2033. https://doi.org/10.14778/3229863.3236252

ADESIT : Visualisez les Limites de vos Données dans un Processus d'Apprentissage Machine

Pierre Faure--Giovagnoli[1,2]
[1]Univ Lyon, INSA Lyon, CNRS, UCBL
LIRIS UMR 5205, Villeurbanne, France
[2]Compagnie Nationale du Rhône
Lyon, France
pierre.faure--giovagnoli@liris.cnrs.fr

Marie Le Guilly
Jean-Marc Petit
Vasile-Marian Scuturici
Univ Lyon, INSA Lyon, CNRS, UCBL
LIRIS UMR 5205, Villeurbanne, France
fisrname.lastname@liris.cnrs.fr

RÉSUMÉ

Grâce aux nombreux outils d'apprentissage machine dont nous disposons aujourd'hui, il est plus facile que jamais de produire un modèle à partir d'un jeu de données dans le cadre d'un problème d'apprentissage supervisé. Cependant, lorsque ce modèle se comporte mal par rapport aux performances attendues, la question sous-jacente de l'existence d'un tel modèle est souvent négligée et l'on peut être tenté de choisir directement une autre architecture ou d'essayer d'autres paramètres d'entraînement. C'est pourquoi la qualité des exemples d'apprentissage doit être prise en compte le plus tôt possible car elle agit comme un go/no-go pour le processus d'apprentissage subséquent qui peut se révéler coûteux.

Avec ADESIT, nous proposons un moyen d'évaluer la capacité d'un jeu de données à donner de bons résultats pour un problème d'apprentissage supervisé à travers des statistiques et une exploration visuelle. Notamment, nous basons notre travail sur des études récentes proposant l'utilisation des dépendances fonctionnelles et spécifiquement l'analyse des contre-exemples. Ainsi, nous fournissons des statistiques sur la propreté du jeu de données mais aussi une limite supérieure théorique sur la précision de prédiction directement liée aux paramètres du problème (incertitude de mesure, généralisation attendue...). En bref, ADESIT est destiné à faire partie d'un processus itératif de raffinement des données, juste après la sélection et juste avant le processus d'apprentissage. Avec une analyse plus poussée pour un problème donné, l'utilisateur peut caractériser, nettoyer et exporter des sous-ensembles sélectionnés dynamiquement, permettant de mieux comprendre quelles régions des données pourraient être affinées et où la précision des données doit être améliorée en utilisant, par exemple, des capteurs nouveaux ou plus précis.

KEYWORDS

contre-exemples, apprentissage machine, visualisation, graphe, données massives, fastg3, modèle

1 INTRODUCTION

Considérons un problème d'apprentissage supervisé (AS) où la tâche est de prédire un attribut cible C à partir d'un ensemble d'attributs X en utilisant un ensemble d'exemples d'apprentissage. Le but est de trouver une fonction f (aussi appelée modèle) telle que $f(X) \simeq C$. Pour ce faire, des exemples d'entraînement sont donnés à un algorithme d'apprentissage pour déduire une telle fonction qui peut ensuite être évaluée sur des exemples de test. Supposons que la précision obtenue soit inférieure aux attentes : *que faut-il remettre en question ?* On pourrait être tenté d'essayer différents paramètres d'apprentissage ou même de modifier l'architecture du modèle. Si cette approche peut fonctionner dans certains cas, remettre en question l'existence même de f devrait également être l'une des principales préoccupations pour éviter un processus d'AS infructueux ou, inversement, pour aider les experts du domaine à faire confiance au modèle obtenu. Dans la pratique, la qualité et l'exhaustivité des exemples d'apprentissage sont parmi les principaux facteurs de réussite de l'AS. Des données bruitées, imprécises ou incomplètes peuvent conduire à des modèles médiocres et il est parfois même difficile de comprendre que les données d'apprentissage elles-mêmes sont à blâmer.

Dans cette étude, nous reprenons des travaux récents proposant l'utilisation des dépendances fonctionnelles (DFs) et spécifiquement l'analyse des contre-exemples pour trouver des contradictions dans un jeu de données. Pour le cas spécifique de l'AS, on comprend intuitivement que l'entraînement à partir d'exemples ayant des causes égales (X) et des cibles différentes (C) est susceptible de poser problème. En tant que fonction, f doit donner une réponse unique pour une entrée donnée. Ce type de contradiction peut être dû à du bruit dans les données (précision du capteur, erreur d'entrée...) mais il peut aussi révéler des attributs manquants ou simplement la nature stochastique du phénomène. Par conséquent, trouver des régions avec une forte densité de contre-exemples est d'un grand intérêt pour le prétraitement des données, pour mieux comprendre les limites du jeu de données brute et de la formulation du problème, mais aussi pour guider les interactions avec les experts du domaine. Si les limites données par l'analyse des contre-exemples ne sont pas adaptées aux besoins métiers, il faut affiner le processus en travaillant sur l'acquisition ou le traitement des données.

Ainsi, nous présentons ADESIT (Advanced Data Exploration and Selection Interactive Tool), un application Web graphique et intuitive permettant d'évaluer les limites d'un jeu de données pour un problème d'AS en collaboration avec les experts métiers. Basée sur les contre-exemples, cette évaluation est faite par des mesures statistiques et une exploration visuelle interactive. Pour un jeu de données, un ensemble d'attributs X et un attribut à prédire C, ADESIT aide l'utilisateur à comprendre le pouvoir prédictif du jeu de données mais aussi les raffinements et améliorations potentiels

dans ses attributs ou sa sélection des données. Nous proposons plusieurs statistiques sur le jeu de données jusqu'à des granularités au niveau du tuple ainsi qu'une représentation interactive des données et des contre-exemples trouvés. Juste avant la création du modèle, ADESIT propose à l'utilisateur de comprendre les limites de son jeu de données, participant ainsi à tout le processus de l'acquisition au prétraitement. Pour un nouveau problème d'AS, nous voulons que l'utilisateur soit rapidement en mesure d'évaluer les performances qu'il ne peut dépasser et les mesures à prendre avant d'aborder le processus d'apprentissage.

2 VUE GÉNÉRALE DU SYSTÈME

2.1 Analyse des contre-exemples

On considère une relation $r[U]$ avec U un ensemble d'attributs et une cible C à prédire à partir d'un ensemble d'attributs X tel que $X \cup \{C\} \subseteq U$. Ainsi, la fonction sous-jacente $f(X) \simeq C$ peut être représentée par une DF sous la forme $\varphi : X \rightarrow C$. Dans notre cas d'étude par exemple, on peut utiliser le débit et l'élévation d'un fleuve pour prédire la puissance d'une turbine tel que $f(debit, elevation) \simeq puissance$ devient $debit, elevation \rightarrow puissance$.

Mesurer la véracité de φ dans r et donc la présence de la fonction f associée permet alors d'évaluer le potentiel d'apprentissage d'un jeu de données. Notamment, ADESIT propose l'étude des contre-exemples : un contre-exemple est une paire de tuples avec des valeurs d'attributs X égales et des cibles C différentes. Dans cette étude, le calcul des contre-exemples est étendu à une classe de DFs non-strictes où l'égalité stricte est remplacée par prédicats permettant de prendre en compte des notions d'incertitudes et de généralisation. Dans ce cas, la paire de tuples (u, v) forme un contre-exemple ssi pour tout $A \in X$, $u[A] \simeq v[A]$ et $u[C] \not\simeq v[C]$ avec $\simeq$ un prédicat lié à une mesure de similarité.

Si visualiser les contre-exemples permet déjà de mieux comprendre ses données, ADESIT propose également 3 indicateurs basés sur les contre-exemples (issus de [2]). Étendus à la famille des DF non-strictes pour cette étude (similaire à [4]), ces indicateurs sont définis comme suit : g_1 la proportion de contre-exemples, g_2 la proportion de tuples impliqués dans au moins un contre-exemple et g_3 la proportion de tuples à enlever de r de sorte à ce qu'il n'y ait plus de contre-exemples. g_3 est notamment une limite supérieure sur la précision de n'importe quel modèle dans le cas des DF strictes [3].

2.2 Aperçu technique

Pour énumérer les contre-exemples, les tuples doivent être comparés deux à deux et sont ajoutés à l'ensemble des contre-exemples s'ils sont en accord sur X et en désaccord sur C. Comme cette opération est quadratique par rapport au nombre de tuples, elle n'est pas facilement adaptable aux grands jeux de données. Néanmoins, cette énumération deux à deux a été largement étudiée pour le *record linkage* et les *similarity joins* où des optimisations ont été proposées. Selon le type d'attribut, des techniques telles que le *blocking* et/ou un algorithme de fenêtre glissante peuvent être utilisées pour des gains remarquables en temps de calcul.

Une fois les contre-exemples trouvés, g_1 est leur nombre et g_2 le nombre de tuples impliqués dans un contre-exemple. En revanche, g_3 nécessite de trouver le nombre minimum de tuples à supprimer de la relation r pour satisfaire $X \rightarrow C$. Si son calcul est trivial dans le cas de DFs classiques [1], l'utilisation de DFs non-strictes le rend NP-Hard (preuve similaire à [4]). De plus, en utilisant une représentation sous forme de graphe où les nœuds correspondent à des tuples et les arêtes relient les nœuds impliqués dans un contre-exemple, le calcul de g_3 devient équivalent au Minimum Vertex Cover, ce qui permet de bénéficier de l'importante littérature sur le sujet. Nous proposons à la fois un algorithme d'approximation et un algorithme exact (pour les petits jeux de données).

2.3 ADESIT

ADESIT est une application Web (adesit.datavalor.com) permettant un collaboration aisée entre les spécialistes en données et les experts métiers. Son interface est présentée en Figure 1.

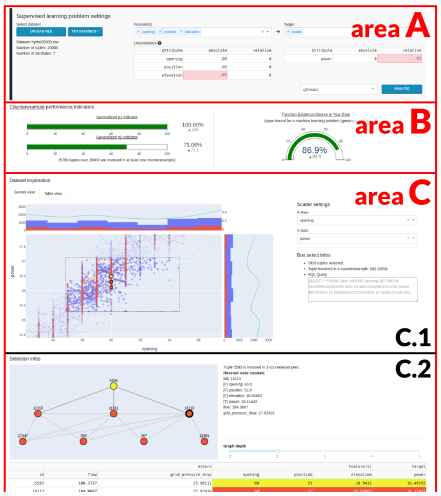


Figure 1: Interface labellée d'ADESIT. A : Paramètre du problème d'apprentissage supervisé. B : Indicateurs de contre-exemples. C : Exploration des contre-exemples.

REMERCIEMENTS

Nous remercions Matteo Dumont et Antoine Mandin pour leur aide sur le développement initial d'ADESIT. Nous remercions également Benjamin Bertin et Vincent Barellon pour avoir testé l'application et leur aide pour son déploiement. Enfin, nous remercions Datavalor de l'INSA Lyon et la Compagnie Nationale du Rhône pour avoir financé une partie de ce travail.

RÉFÉRENCES

[1] Yka Huhtala, Juha Kärkkäinen, Pasi Porkka, and Hannu Toivonen. 1999. TANE : An efficient algorithm for discovering functional and approximate dependencies. *The computer journal* 42, 2 (1999), 100–111.

[2] Jyrki Kivinen and Heikki Mannila. 1995. Approximate inference of functional dependencies from relations. *Theoretical Computer Science* 149, 1 (1995), 129 – 149. Fourth International Conference on Database Theory (ICDT '92).

[3] Marie Le Guilly, Jean-Marc Petit, and Vasile-Marian Scuturici. 2020. Evaluating Classification Feasibility Using Functional Dependencies. *Trans. Large Scale Data Knowl. Centered Syst.* 44 (2020), 132–159.

[4] Shaoxu Song. 2010. *Data dependencies in the presence of difference.* Ph. D. Dissertation. Hong Kong University of Science and Technology.

A Demonstration of the Exathlon Benchmarking Platform for Explainable Anomaly Detection*

Vincent Jacob
vincent.jacob@polytechnique.edu
Ecole Polytechnique
France

Fei Song
fei.song@polytechnique.edu
Ecole Polytechnique
France

Arnaud Stiegler
arnaud.stiegler@polytechnique.edu
Ecole Polytechnique
France

Bijan Rad
bijan.rad@polytechnique.edu
Ecole Polytechnique
France

Yanlei Diao
yanlei.diao@polytechnique.edu
Ecole Polytechnique
France

Nesime Tatbul
tatbul@csail.mit.edu
Intel Labs and MIT
USA

Access to high-quality data repositories and benchmarks have been instrumental in advancing the state of the art in many experimental research domains. While advanced analytics tasks over time series data have been gaining lots of attention, lack of such community resources severely limits scientific progress. In this demonstration, we showcase Exathlon, the first comprehensive public benchmark for explainable anomaly detection over high-dimensional time series data. Exathlon has been systematically constructed based on real data traces from repeated executions of large-scale stream processing jobs on an Apache Spark cluster. Some of these executions were intentionally disturbed by introducing instances of six different types of anomalous events (e.g., misbehaving inputs, resource contention, process failures). For each of the anomaly instances, ground truth labels for the root cause interval as well as those for the extended effect interval are provided, supporting the development and evaluation of a wide range of anomaly detection (AD) and explanation discovery (ED) tasks. This demonstration presents Exathlon's curated anomaly dataset, novel benchmarking methodology, and end-to-end data science pipeline in action via example usage scenarios. The dataset, code, and documentation for Exathlon are publicly available at https://github.com/exathlonbenchmark/exathlon.

*This paper was previously published in the Demonstration Track of the 47th International Conference on Very Large Data Bases (VLDB 2021).

Julien Aimonier-Davat, Hala Skaf-Molli, Pascal Molli SaGe-Path : Pay-as-you-go SPARQL Property Path Queries processing using Web Preemption

SaGe-Path : Pay-as-you-go SPARQL Property Path Queries Processing using Web Preemption

Julien Aimonier-Davat
Université de Nantes, LS2N
Nantes, France
julien.aimonier-davat@univ-nantes.fr

Hala Skaf-Molli
Université de Nantes, LS2N
Nantes, France
hala.skaf@univ-nantes.fr

Pascal Molli
Université de Nantes, LS2N
Nantes, France
pascal.molli@univ-nantes.fr

RÉSUMÉ

Les requêtes SPARQL property path représentent un outil indispensable pour chercher des motifs complexes dans un graphe de connaissance. Cependant, évaluer ces requêtes sur des données en ligne, et obtenir des résultats complets, est difficile. Du fait de leur complexité, les requêtes property path sont souvent intérrompues par les politiques d'usage équitable en place sur les SPARQL endpoints. Afin de garantir des résultats complets, SaGe-Path repose sur la préemption Web et le concept de fermetures transitives partielles (PTC). Lors de l'évaluation d'une expression property path, le serveur PTC limite l'exploration du graphe à une profondeur k, définie à l'avance. Les noeuds visités à une profondeur k sont appelés noeuds frontières et sont envoyés au client. En utilisant ces noeuds frontières, le client PTC est alors capable de générer de nouvelles requêtes, afin de continuer l'exploration du graphe. Ainsi, SaGe-Path propose une approche *pay-as-you-go* qui permet d'évaluer efficacement des requêtes SPARQL property path sur des données en ligne, avec la garantie d'obtenir des résultats complets. L'objectif de cette démonstration est de montrer comment des requêtes qui ne terminent pas sur Wikidata, un SPARQL endpoint publique, terminent en utilisant SaGe-Path. Une interface utilisateur permet de suivre l'évaluation d'une requête en temps réel, afin de comprendre les overheads de l'approche et l'impact des différents paramètres sur l'exécution de la requête.

Tell Me What Air You Breathe, I Tell You Where You Are

Hafsa El Hafyani
DAVID Lab
UVSQ - Université Paris-Saclay
Versailles, France
hafsa.el-hafyani@uvsq.fr

Mohammad Abboud
DAVID Lab
UVSQ - Université Paris-Saclay
Versailles, France
mohammad.abboud@uvsq.fr

Jingwei Zuo
DAVID Lab
UVSQ - Université Paris-Saclay
Versailles, France
jingwei.zuo@uvsq.fr

Karine Zeitouni
DAVID Lab
UVSQ - Université Paris-Saclay
Versailles, France
karine.zeitouni@uvsq.fr

Yehia Taher
DAVID Lab
UVSQ - Université Paris-Saclay
Versailles, France
yehia.taher@uvsq.fr

ABSTRACT

Wide spread use of sensors and mobile devices along with the new paradigm of Mobile Crowd-Sensing (MCS), allows monitoring air pollution in urban areas. Several measurements are collected, such as Particulate Matters, Nitrogen dioxide, and others. Mining the context of MCS data in such domains is a key factor for identifying the individuals' exposure to air pollution, but it is challenging due to the lack or the weakness of predictors. We have previously developed a multi-view learning approach which learns the context solely from the sensor measurements. In this demonstration, we propose a visualization tool (**COMIC**) showing the different recognized contexts using an improved version of our algorithm. We also demonstrate the change points detected by a multi-dimensional CPD model. We leverage real data from a MCS campaign, and compare different methods.

KEYWORDS

Activity Recognition, Multivariate Time Series Classification, Multi-view Learning, Mobile Crowd Sensing, Air Quality Monitoring

1 COMIC OVERVIEW

Air quality and exposure to pollution is a central concern for people living in urban areas. As the harmful effects of air pollutants on their health is alarming. The key concern to reduce the risk of these pollutants on individual's health is by understanding the totality of exposure. Air pollution monitoring is getting more interest nowadays, due to the rapid advances of the Internet of things (IoT) along with the emergence of the Mobile Crowd Sensing (MCS) paradigm.

The mentioned technologies coupled with the widespread use of GPS, allows volunteers to contribute their collected data in order to get personalized insights about their exposures to pollution. Polluscope [1] is a French project deployed in Île-de-France (i.e., Paris region), and is a typical use case study based on MCS. In Polluscope, participants are equipped with a sensor kit which can measure different pollutants such as Nitrogen dioxide (NO2), Particulate Matters (PMS), Black Carbon (BC), Temperature, etc… independently form their environment either indoor or outdoor.

Air quality strongly depends on the context of the participant, thus in order to understand and identify participants' exposure to pollution, it is essential to identify the context of the participants. To avoid miss-classification of the exposure wrt the context (micro-environment), the participants need to fill a time-use diary, but in real-life, they rarely thoroughly do this self-reporting task. Therefore, mining the context of participants based on the data collected from the crowd is an attractive solution. But, it is challenging due to the imperfection of the data as shown in [6]. Consequently, learning from individual sensor data fails to identify the micro-environment. This classification might be improved by combining multiple sensor data. However, this is not straightforward due to the complexity and interdependence of such multi-dimensional time series [3].

This task is more or less related to human activity recognition, where there exists a long established research, and a wide range of applications. Different types of data are used ranging from GPS only, to one or many accelerometers, sound or combinations. An attempt to work with environmental data was introduced by Asimina et al. [2] where the authors explore the capability of predicting location from sensors data using an Artificial Neural Network (ANN) model. The authors use low-cost individual sensors and GPS for data collection, and provide a location predictive model based on Artificial Neural Network (ANN) to derive the time-space activity daily profile of the individuals. Although their approach predicts very well indoor locations, however, there is still room for improvement in discriminating between in transit and outdoor locations.

Furthermore, there is a need for a full-fledged implementation that can be used in real-world applications to fill the gap between data collection and context recognition using environmental data. In our case, the input is both GPS and environmental sensor data. We formulate the task of recognition of micro-environment as a multivariate time series classification (MTSC) [5]. In a previous work [1], we've proposed a multi-view stacking generalization approach in order to detect the micro-environments from the air pollution data collected based on different learners.

In this demonstration, we include time series segmentation based on change point detection to emphasize the automatic detection of

[1] http://polluscope.uvsq.fr

the change in the user's context. Furthermore, we develop a visualization tool **COMIC** (**C**ontext **O**f **M**ob**I**le **C**rowdsensing) showing the different recognized contexts and illustrating the importance of multi-view approach when compared to single view approach and other baseline learners. This visualization interface highlights also the importance of our approach vis-a-vis users' declared contexts.

For more information, refer to our paper in [4], and the presentation here (or in this URL: https://youtu.be/jgWPlxJ4-ms).

ACKNOWLEDGMENTS

This work has supported by the French National Research Agency (ANR) project Polluscope, funded under the grant agreement ANR-15-CE22-0018, by the H2020 EU GO GREEN ROUTES funded under the research and innovation programme H2020- EU.3.5.2 grant agreement No 869764, and by the DATAIA convergence institute project StreamOps, as part of the Programme d' Investissement d'Avenir, ANR-17-CONV-0003. Part of the equipment was funded by iDEX Paris-Saclay, in the framework of the IRS project ACE-ICSEN, and by the Communauté d'agglomération Versailles Grand Parc. We are thankful to all the members of the Polluscope consortia who contributed in one way or another to this work: Salim Srairi and Jean-Marc Naude (CEREMA) who conducted the campaign; Boris Dessimond and Isabella Annesi-Maesano (Sorbonne University) for their contribution to the campaign; Valerie Gros, Baptiste Languille and Nicolas Bonnaire (LSCE), and Anne Kauffmann and Christophe Debert (Airparif) for their contribution in the periodic qualification of the sensors and their feedback. Finally, we would like to thank the participants for their involvement.

REFERENCES

[1] Mohammad Abboud, Hafsa El Hafyani, Jingwei Zuo, Karine Zeitouni, and Yehia Taher. 2021. Micro-environment Recognition in the context of Environmental Crowdsensing. *Proceedings of the Workshops of the EDBT/ICDT 2021 Joint Conference* 2841 (2021).

[2] Stamatelopoulou Asimina, D. Chapizanis, S. Karakitsios, P. Kontoroupis, D. Asimakopoulos, T. Maggos, and D. Sarigiannis. 2018. Assessing and enhancing the utility of low-cost activity and location sensors for exposure studies. *Environmental Monitoring and Assessment* 190 (2018), 1–12.

[3] Hafsa El Hafyani. 2020. Big Data Series Analytics in the Context of Environmental Crowd Sensing. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 246–247.

[4] Hafsa El Hafyani, Mohammad Abboud, Jingwei Zuo, Karine Zeitouni, and Yehia Taher. 2021. *Tell Me What Air You Breath, I Tell You Where You Are.* Association for Computing Machinery, New York, NY, USA, 161–165. https://doi.org/10.1145/3469830.3470914

[5] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery* 33, 4 (2019), 917–963.

[6] Baptiste Languille, Valérie Gros, Nicolas Bonnaire, Clément Pommier, Cécile Honoré, Christophe Debert, Laurent Gauvin, Salim Srairi, Isabella Annesi-Maesano, Basile Chaix, et al. 2020. A methodology for the characterization of portable sensors for air quality measure with the goal of deployment in citizen science. *Science of the Total Environment* 708 (2020), 134698.

2

7 Résumés des articles de doctorant

Enabling Reproducible Analysis of Complex Workflows on the Edge-to-Cloud Continuum

Daniel Rosendo
daniel.rosendo@inria.fr
Univ Rennes, INSA Rennes, CNRS, Inria, IRISA
Rennes, France

Alexandru Costan
alexandru.costan@inria.fr
Univ Rennes, INSA Rennes, CNRS, Inria, IRISA
Rennes, France

Gabriel Antoniu
gabriel.antoniu@inria.fr
Univ Rennes, CNRS, Inria, IRISA
Rennes, France

Patrick Valduriez
patrick.valduriez@inria.fr
Univ Montpellier, Inria, CNRS, LIRMM
Montpellier, France

ABSTRACT

Distributed digital infrastructures for computation and analytics are now evolving towards an interconnected ecosystem allowing complex applications to be executed from IoT Edge devices to the HPC Cloud (aka the *Computing Continuum*, the *Digital Continuum*, or the *Transcontinuum*). Understanding end-to-end performance in such a complex continuum is challenging. This breaks down to reconciling many, typically contradicting application requirements and constraints with low-level infrastructure design choices. One important challenge is to accurately reproduce relevant behaviors of a given application workflow and representative settings of the physical infrastructure underlying this complex continuum. We introduce a rigorous methodology for such a process and validate it through **E2C***lab*. It is the first platform to support the complete experimental cycle across the Computing Continuum: deployment, analysis, optimization. Preliminary results with real-life use cases show that **E2C***lab* allows one to understand and improve performance, by correlating it to the parameter settings, the resource usage and the specifics of the underlying infrastructure.

CCS CONCEPTS

• **Computing methodologies** → **Distributed computing methodologies**; **Machine learning**; • **General and reference** → **Experimentation**; **Measurement**;

KEYWORDS

Methodology, Computing Continuum, Reproducibility, Machine Learning, Optimization

1 CONTEXT

The explosion of data generated from the Internet of Things (IoT) and the need for real-time analytics has resulted in a shift of the data processing paradigms towards decentralized and multi-tier computing infrastructures and services. New challenging application scenarios are emerging from a variety of domains such as healthcare, self-driving vehicles, precision agriculture, *etc.* This contributes to the emergence of what is called the *Computing Continuum* [2]. It seamlessly combines resources and services at the center (e.g., in Cloud datacenters), at the Edge, and in-transit, along the data path. Typically data is first generated and preprocessed (e.g., filtering, basic inference) on Edge devices, while Fog nodes further process partially aggregated data. Then, if required, data is transferred to HPC-enabled Clouds for Big Data analytics, AI model training, and global simulations.

2 PROBLEM STATEMENT

Despite an always increasing number of dedicated systems for data processing on each component of the continuum, this vision of ubiquitous computing remains largely unrealized. This is due to the complexity of deploying large-scale, real-life applications on such heterogeneous infrastructures, which breaks down to configuring a myriad of system-specific parameters and reconciling many requirements or constraints, e.g., in terms of communication latency, energy consumption, resource usage, data privacy.

A first step towards reducing this complexity and enabling the Computing Continuum vision is to enable a **holistic understanding of performance** in such environments. That is, finding a rigurous approach to answering questions like: *(1) Which system parameters and infrastructure configurations impact on performance and how? (2) Where should the workflow components be executed to minimize communication costs and end-to-end latency?*

3 STATE OF THE ART

Approaches based on workflow modelling [6] and simulation [7] raise some important challenges in terms of specification, modelling, and validation in the context of the Computing Continuum. For example, it is increasingly difficult to assess the impact of the inherent complexity of hybrid Edge-Cloud deployments on performance. At this stage, experimental evaluation remains the main approach to gain accurate insights of performance metrics and to build precise approximations of the expected behavior of large-scale applications on the Computing Continuum, as a first step prior to modelling.

4 CHALLENGES

A key challenge in this context is to be able to **reproduce in a representative way the application behavior in a controlled**

environment, for extensive experiments in a large-enough spectrum of potential configurations of the underlying Edge-Fog-Cloud infrastructure. In particular, this means rigorously mapping the scenario characteristics to the *experimental environment*, identifying and controlling the relevant *configuration parameters* of applications and system components, defining the relevant *performance metrics*. The above process is non-trivial due to the multiple combination possibilities of heterogeneous hardware/software resources, system components for data processing, analytics or model training.

5 PHD OBJECTIVES

In order to allow other researchers to leverage the experimental results and advance knowledge in different domains, experimental methodologies need to enable three R's of research quality: **Repeatability, Replicability,** and **Reproducibility (3R's)**. This translates to establishing a *well-defined experimentation methodology* and providing *transparent access to the experiment artifacts* and *experiment results*.

The Computing Continuum vision calls for a rigorous and systematic methodology to map real-world application components and dependencies to infrastructure resources, a complex process that can be error prone. Key research goals are: *1)* to identify relevant characteristics of the application workloads and of the underlying infrastructure as a means to enable accurate experimentation and benchmarking in relevant infrastructure settings in order to understand their performance; and *2)* to ensure research quality aspects such as the 3R's.

6 OUR CONTRIBUTION: E2C*LAB*

E2C*lab* [5] implements a methodology that supports the complete experimental cycle across the edge-to-cloud continuum, including deployment, configuration, optimization, and experiment execution in a reproducible way. It may be used by researches to deploy real-life applications on large-scale testbeds and perform meaningful experiments in a systematic manner. The **main contributions** of this work are:

A **rigorous methodology for designing experiments with real-world workloads on the Computing Continuum** spanning from the Edge to the Cloud; this methodology provides guidelines to move from real-world use cases to the design of relevant testbed setups for experiments enabling researchers to understand performance and to ensure the 3R's properties.

A novel **framework named E2C***lab* that implements this methodology and allows researchers to deploy their use cases on real-world large-scale testbeds, e.g., G5k [1]. To the best of our knowledge, **E2C***lab* is the first platform to support the complete analysis cycle of an application on the Computing Continuum: *(i)* the configuration of the experimental environment; *(ii)* the mapping between the application parts and machines on the Edge, Fog and Cloud; *(iii)* the deployment and monitoring of the application on the infrastructure; and *(iv)* the automated execution and gathering of results.

A **large scale experimental validation** on the G5K [1] testbed with Pl@ntNet [3], a real-life use case. **E2C***lab* allows optimizing the Pl@ntNet's performance based on the analysis of the parameter settings and correlation to processing time and resource usage [4].

7 PRELIMINARY RESULTS

We illustrate [5] **E2C***lab* usage with a **real-life Smart Surveillance System** deployed on the Grid'5000 testbed, showing that our framework allows one to understand how the Cloud-centric and the hybrid Edge-Cloud processing approaches impact performance metrics such as latency and throughput.

Besides, we validate [4] **E2C***lab* with **Pl@ntNet**, another **real-life use case**. We demonstrate that **E2C***lab* guides on the optimization of the Pl@ntNet performance based on the analysis of the parameter settings and correlation to processing time and resource usage. Preliminary results show that Pl@ntNet's deployment configurations found by **E2C***lab* perform better than the current ones used in the production servers.

8 NEXT RESEARCH STEPS

We are exploring **parallel and scalable optimization** techniques that supports surrogate modeling optimization for large-scale multi-objective optimization problems. In this direction, we have an ongoing collaboration with Argonne National Laboratory members, where we are discussing potential solutions to support the optimization of complex application workflows on the Edge-to-Cloud Continuum.

Furthermore, since **E2C***lab* supports **reproducible experiments**, we will explore and propose techniques for **runtime provenance** collection in large-scale and distributed experimental environments. The goal is to provide additional context that more accurately explains the experiment execution and results. This research direction is a collaboration with the Federal University of Rio de Janeiro, Brazil.

REFERENCES

[1] Raphaël Bolze, Franck Cappello, Eddy Caron, Michel Dayde, Frédéric Desprez, Emmanuel Jeannot, Yvon Jégou, Stephane Lanteri, Julien Leduc, Nouredine Melab, Guillaume Mornet, Raymond Namyst, Pascale Primet, Benjamin Quétier, Olivier Richard, El-Ghazali Talbi, and Iréa Touche. 2006. Grid'5000: A Large Scale And Highly Reconfigurable Experimental Grid Testbed. *International Journal of High Performance Computing Applications* 20, 4 (2006), 481–494. https://doi.org/10.1177/1094342006070078

[2] ETP4HPC. 2020. *ETP4HPC Strategic Research Agenda.* Retrieved April 29, 2020 from https://www.etp4hpc.eu/sra.html

[3] Alexis Joly, Pierre Bonnet, Hervé Goëau, Julien Barbe, Souheil Selmi, Julien Champ, Samuel Dufour-Kowalski, Antoine Affouard, Jennifer Carré, Jean-François Molino, et al. 2016. A look inside the Pl@ntNet experience. *Multimedia Systems 22*, 6 (2016), 751–766.

[4] Daniel Rosendo, Alexandru Costan, Gabriel Antoniu, Matthieu Simonin, Jean-Christophe Lombardo, Alexis Joly, and Patrick Valduriez. 2021. Reproducible performance optimization of complex applications on the edge-to-cloud continuum. *arXiv preprint arXiv:2108.04033* (2021).

[5] Daniel Rosendo, Pedro Silva, Matthieu Simonin, Alexandru Costan, and Gabriel Antoniu. 2020. E2Clab: Exploring the Computing Continuum through Repeatable, Replicable and Reproducible Edge-to-Cloud Experiments. In *2020 IEEE International Conference on Cluster Computing (CLUSTER).* IEEE, 176–186.

[6] Shazia Sadiq, Maria Orlowska, Wasim Sadiq, and Cameron Foulger. 2004. Data Flow and Validation in Workflow Modelling. In *Proceedings of the 15th Australasian database conference-Volume 27.* 207–214.

[7] Sergej Svorobej, Patricia Takako Endo, Malika Bendechache, Christos Filelis-Papadopoulos, Konstantinos M Giannoutakis, George A Gravvanis, Dimitrios Tzovaras, James Byrne, and Theo Lynn. 2019. Simulating Fog and Edge Computing Scenarios: An Overview and Research Challenges. *Future Internet* 11, 3 (2019), 55.

Deploying Heterogeneity-aware Deep Learning Workloads on the Computing Continuum

Thomas Bouvier
thomas.bouvier@irisa.fr
Univ Rennes, CNRS, Inria, IRISA
Rennes, France

Alexandru Costan
alexandru.costan@irisa.fr
Univ Rennes, INSA Rennes, CNRS, Inria, IRISA
Rennes, France

Gabriel Antoniu
gabriel.antoniu@irisa.fr
Univ Rennes, CNRS, Inria, IRISA
Rennes, France

ABSTRACT

The increasing need for real-time analytics motivated the emergence of new incremental methods to learn representations from continuous flows of data, especially in the context of the Internet of Things. This trend led to the evolution of centralized computing infrastructures towards interconnected processing units spanning from edge devices to cloud data centers. This new paradigm is referred to as the Computing or Edge-to-Cloud Continuum. However, the network and compute heterogeneity across and within clusters may negatively impact Deep Learning (DL) training. We introduce a roadmap for understanding the end-to-end performance of DL workloads in such heterogeneous settings. The goal is to identify key parameters leading to stragglers and devise novel intra- and inter- cluster strategies to address them. We will explore various policies aiming to improve makespan, cost and fairness objectives while ensuring system scalability.

CCS CONCEPTS

• **Computer systems organization** → **Grid computing**; • **Computing methodologies** → **Distributed computing methodologies**; **Machine learning**.

KEYWORDS

Deep Learning, incremental learning, heterogeneous systems, Computing Continuum.

1 CONTEXT

State-of-the-art **deep learning** (DL) models outperform human experts' capacity in many domains, including image classification, machine translation or gaming. With the recent rise of the Internet of Things (IoT), input data is generated at an increasingly rapid pace by sensors all over the globe. This trend motivated the emergence of the **Computing Continuum** as a means to distribute computation from centralized clouds towards multi-tier processing units (*i.e.* mini-clusters in the fog) or edge devices themselves. Such infrastructure aims to optimize the performance of geo-distributed applications, benefiting from data locality to decrease communication costs and latency. However, **network and compute heterogeneity** should be carefully considered to efficiently leverage this continuum. Besides, deep neural networks (DNNs) need to shift from one-time training (typically used for static data) to approaches capable of learning from incoming flows of data. **Incremental learning** is such an emerging DL method where progressively available data is used to extend the model's knowledge.

2 CHALLENGES

Distributing DL training across fog mini-clusters and cloud data centers poses several challenges:

Network heterogeneity. Distributed DNNs suffer from low-speed and high-latency wide-area networks (WANs) connecting distant nodes. These heterogeneous links lead to stragglers, making it challenging for DL algorithms to perform efficient training.

Compute heterogeneity. As Moore's law came to an end, GPUs and other specialized accelerators emerged alongside traditional CPUs to train DNNs. In practice, clusters provide a different number of nodes equipped with different hardware or virtualized resources, leading to heterogeneous performance across and within clusters.

Scale. By nature, edge-to-cloud computing is geo-distributed. At scale, this characteristic aggravates network heterogeneity, leading to communication bottlenecks on multiple low-bandwidth nodes.

3 PROBLEM STATEMENT

Conciliating both network and compute heterogeneity to distribute the DL training efficiently in large-scale scenarios is therefore a problem gaining more and more interest from both the DL and the HPC – Big Data communities. Strategies at system and algorithmic levels should be explored as users might want to optimize different objectives based on their high-level goals, such as **minimizing the time to complete the job (makespan), costs, or improving fairness**.

4 PHD OBJECTIVES

During my PhD (started in 2021), I will study the following research questions: *1) How much can one improve (or degrade) the efficiency of DL training by performing it in the fog (closer to the edge) rather than performing it in the cloud?* and *2) How to account for the heterogeneous network and compute capabilities of the processing units across the Computing Continuum?*

In order to answer these questions, I plan to devise a set of **strategies and algorithms optimizing DL workloads** in heterogeneous settings. More specifically, in order to assess their impact, I plan to study how the proposed techniques affect the **accuracy** in classifying new data and to what extent they apply to **incremental learning**.

5 STATE OF THE ART

Parallel stochastic gradient descent (PSGD) and its variants have been widely used to train large DNNs. Current approaches allow training model subsets on their local data asynchronously, and exchange gradients via peer-to-peer communication to synchronize local subsets. Techniques like bounded staleness, backup workers and synchronization queues [4] have been devised to account for dynamic network heterogeneity, whereas the selection of the fastest links [2, 3], prevent the emergence of deterministic stragglers. Skipping SGD iterations and optimizing the partitioning of input data [1] help with deterministic compute heterogeneity. At the cluster level, heterogeneity-aware schedulers [5] allow sustaining heavier workloads than agnostic ones.

However, these approaches do not comprehensively consider the challenges posed by the continuum. In particular, they enable DL deployments on distributed platforms by addressing the issues of network and compute heterogeneity separately.

6 CONTRIBUTIONS

A first step towards executing efficient DL training in heterogeneous settings is to get a **holistic understanding of performance**. For this purpose, we will leverage E2Clab [6], a framework supporting the complete experimental cycle across the continuum, including configuration, deployment and execution in a reproducible way. We will extend it to **identify inefficient gradients computation and propagation** across nodes.

The next step is to devise novel intra- and inter- cluster strategies to address the above observations. In particular, the following facets should be considered:

- **Coarse-grained compute capacity**. The compute capacity of clusters should be estimated to adjust the initial data parallelism batch size accordingly. Input data migration could be performed at runtime.
- **Fine-grained compute capacity**. DNN training proceeds in iterations. Within a cluster, heterogeneity-aware scheduling strategies should improve utilization and optimize fairness by time-multiplexing jobs over CPUs and GPUs.
- **Deterministic network capacity**. Fast links should be favored to exchange data. Limiting communication to certain links requires ensuring the convergence of the SGD algorithm across nodes.
- **Dynamic network capacity**. Temporary network slowdowns should be detected in real time and countered by decentralized asynchronous SGD methods.

The main contribution of my PhD will be **the design of a prototype framework helping to distribute DL training** (Figure 1), accounting for deterministic and dynamic of both network and compute heterogeneity.

7 METHODOLOGY & ROADMAP

During the first year, I plan to conduct experiments on Grid'5000, a large-scale testbed for distributed computing. The methodology implemented in E2Clab will help reproduce both relevant behaviors of the given DL workloads and representative settings of the physical infrastructure underlying the continuum. The goal is to identify the main factors leading to stragglers slowing down the whole training process. The architectural differences of DNNs should be compared.

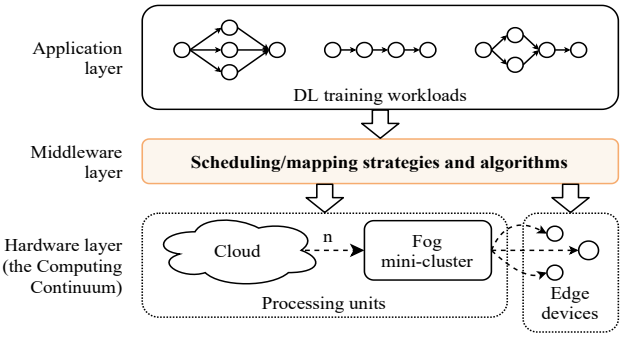


Figure 1: Processing DL workloads on the Computing Continuum.

The outcomes will drive the design of a framework mitigating the identified infrastructure bottlenecks in DL scenarios. By explicitly considering heterogeneity, various policies should improve the following objectives:

- **Makespan** refers to completing the training process as soon as possible.
- **Cost** is relevant when using elastic resources from public clouds. It specifically aims to minimize communication and resource usage costs.
- **Fairness** aims to optimize the fair sharing of processing units across the continuum.

Eventually, it will be interesting to study the extent to which these strategies apply to incremental learning. The makespan should be largely improved by deferring the global model synchronization across more distant nodes.

REFERENCES

[1] Rankyung Hong and Abhishek Chandra. 2020. DLion: Decentralized Distributed Deep Learning in Micro-clouds. In *Proceedings of the 30th International Symposium on High-Performance Parallel and Distributed Computing*. 227–238.

[2] Kevin Hsieh, Aaron Harlap, Nandita Vijaykumar, Dimitris Konomis, Gregory R Ganger, Phillip B Gibbons, and Onur Mutlu. 2017. Gaia: Geo-distributed Machine Learning Approaching LAN Speeds. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*. 629–647.

[3] Qinyi Luo, Jiaao He, Youwei Zhuo, and Xuehai Qian. 2020. Prague: High-performance Heterogeneity-aware Asynchronous Decentralized Training. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*. 401–416.

[4] Qinyi Luo, Jinkun Lin, Youwei Zhuo, and Xuehai Qian. 2019. Hop: Heterogeneity-aware Decentralized Training. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. 893–907.

[5] Deepak Narayanan, Keshav Santhanam, Fiodar Kazhamiaka, Amar Phanishayee, and Matei Zaharia. 2020. Heterogeneity-aware Cluster Scheduling Policies for Deep Learning Workloads. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*. 481–498.

[6] Daniel Rosendo, Pedro Silva, Matthieu Simonin, Alexandru Costan, and Gabriel Antoniu. 2020. E2Clab: Exploring the Computing Continuum Through Repeatable, Replicable and Reproducible Edge-to-cloud Experiments. In *2020 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 176–186.

Towards designing a temporal graph management system

Maria Massri
IRISA
Rennes, France,
Orange Labs
Cesson-Sévigné, France

ABSTRACT

Temporal graphs are useful for modelling the dynamic behavior of relationship-centered domains. Analyzing these graphs paves the way to the extraction of meaningful insights such as detecting anomalies or forecasting future behavior of underlying systems. In this context, the need of a temporal graph management system that handles not only the structural dimension but also the temporal one becomes highlighted. However, designing such a system imposes a number of challenges such as the storage and querying of temporal graphs which we mainly address in this PhD thesis.

KEYWORDS

Temporal Graphs, Graph management systems, Graph querying language

1 INTRODUCTION

Graphs are used in a myriad of application domains to model relationship-centered data. In this context, Thing'in [1] is a platform that uses a graph to model the connections between connected (smoke detector, cameras, printers, etc.) and non-connected (rooms, tables, roads, etc.) objects. However, smart devices embed a highly dynamic behavior leaving the graph of Thing'in to be naturally modeled as a temporal graph. Indeed, analyzing such a dynamic behavior opens the way to promising applications such as the prevention of critical situations and forecast of future behaviors. In order to keep pace with this demand, a graph management system with an optimized time-version support is needed. For argumentation sake, we showcase in the following example the advantages of a temporal graph management system as compared to conventional (non-temporal) systems.

Example 1.1. A motivational use case is that of smart transportation networks. Figure 1 shows a toy graph modelling the connections of transportation network. In such a graph vertices model smart vehicles, radars, traffic lights, etc. and edges represent the temporary connections between vehicles and their surrounding such as connected sensors, objects connected to a road (cameras, traffic lights, etc.). A temporal graph management system should be able to query not only the structural dimension of underlying graphs but also the temporal one. For instance, non-temporal queries permit to extract: Sensors that are connected to a vehicle, Radars that are connected to a road or owners of a vehicle. However, adding the temporal dimension enrich the expressiveness of queries. That is, it permits the expression of the following queries:

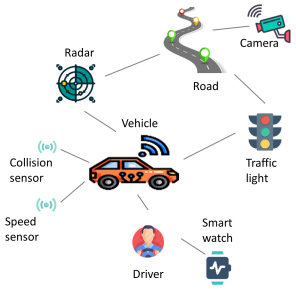


Figure 1: Graph modeling the use case of smart transportation networks

- Q_1: Neighborhood of a car or health status (monitored by smart watch) right before an accident.
- Q_2: Speed recorded by a radar when the latter has detected a certain vehicle.
- Q_3: Average waiting time of a vehicle at a traffic light.
- Q_4: Time during which a sensor connected to a car indicates a value that is higher than a threshold knowing that the vehicle was turned on for less than an hour.

Regarding the additional expressiveness of temporal graph queries, particularly in the analysis of IoT domains, our main motivation is the design and integration of a temporal graph management system into the Thing'in platform. Towards accomplishing this goal, we mainly tackled two challenges that are: the storage techniques and querying languages of temporal property graphs that will be discussed more in details in sections 2 and 3, respectively.

2 CLOCK-G

In this section, we present Clock-G: a temporal graph management system implementing a space-efficient storage technique. A key concept in the design of such a system is the data model. Hence, we have formally defined a temporal property graph as a collection of vertices, edges and values of properties having each a sequence of validity intervals. To clarify, a validity interval refers to the time interval during which an entity was valid. Regarding the outgrowing size of temporal graphs, a crucial consideration is the storage technique of such graphs. One of the proposed methods is the Copy+Log [3, 4, 7] that consists of storing graph updates in temporally disjoint chunks known as time windows along with valid states of the graph known as snapshots. These snapshots represent a materialization

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

of the graph states that are valid between the boundaries of two successive time windows. The advantages of this solution is that recovering the state of the graph at a given time instant requires at most reading a snapshot and the graph updates contained in a single time window. However, snapshots are space consuming because they materialize the state of every existing vertex or edge at a single time instant. This problem becomes even more exacerbated in case of growth-mostly graphs where vertices and edges tend to survive for longer duration and hence are copied over and over again in several snapshots. To overcome this limitation, we proposed the δ-Copy+Log storage approach, then developed Clock-G [5], a graph management system that adopts this method. Our solution mainly differentiates from the Copy+Log method by storing the differences between successive snapshots instead of full snapshots. A crucial concept here is that we store a snapshot after a number of time windows in order to serve as a starting point for query evaluation. Having this, we store graph operations in consecutive time buckets containing each a number M of time windows such that the first $M-1$ time windows ends with a delta, whereas the final time window end with a snapshot. A critical optimization is the forward and backward data representation. That is, half of the deltas and time windows in a bucket is constructed in a forward fashion whereas the other half is constructed in a backward fashion. The rationale behind this choice is the acceleration of the query's execution time. That is, we choose the closest snapshot from which to start the search then compute the result in a forward or backward fashion whether the time instant of that snapshot is lower or greater than the requested one. Knowing that storing deltas adds a querying time overhead, Clock-G offers optimization techniques to mitigate this additional cost. To validate the efficiency of the proposed methods, we evaluated the performance of Clock-G with the use of real and synthetic datasets. Our experimental results demonstrated that the δ-Copy+Log method can significantly reduce the space usage while adding only a negligible overhead to the query execution time as compared to the Copy+Log technique.

3 T-CYPHER

Another essential consideration in the design of a temporal graph management system is the querying language. Indeed, we need to couple Clock-G with a querying language that supports the additional temporal dimension. In this context, we proposed T-Cypher [6]: a time-extended version of OpenCypher [2]. We resume our main contributions as follows: Formalizing the definition of a temporal property graph pattern, extending the syntax of Cypher queries with temporal constructs and formalizing the semantics of the proposed language. That is, we defined a temporal property graph pattern (TPGP) as a graph pattern augmented with temporal variables and temporal constraints. For instance, conventional graph patterns include variables referring to vertices, edges or property values. We extended these patterns with temporal variables on top of which we defined temporal constraints that can be useful for a wide variety of temporal queries. To express TPGP queries, we extended the syntax of Cypher with temporal constructs such as temporal clauses, operators and functions. The extended syntax is non-intrusive in the sense that it offers a straightforward transition for practitioners who are already familiar with the Cypher language.

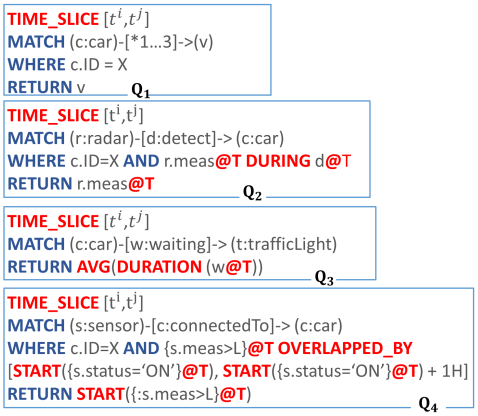


Figure 2: Queries Q_1, Q_2, Q_3 and Q_4 of Example 1.1 expressed with the T-Cypher syntax

Finally, we followed an approach that is similar to the one used in [1] to provide the formal semantics of the extended language. We present in Figure 1.1 how to express queries Q_1, Q_2, Q_3 and Q_4 with the T-Cypher language.

4 SUMMARY AND PERSPECTIVE

To summarize, we proposed a space-efficient technique to store temporal graphs that we have adopted in the design of the temporal graph management system Clock-G. Besides, we proposed and formalized a querying language for temporal property graphs. Our perspective is to couple this querying language with Clock-G by designing a query planner that is optimized for the evaluation of T-Cypher queries.

REFERENCES

[1] Nadime Francis, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, Martin Schuster, Petra Selmer, and Andrés Taylor. 2018. Formal Semantics of the Language Cypher. *CoRR* abs/1802.09984 (2018). arXiv:1802.09984 http://arxiv.org/abs/1802.09984

[2] Nadime Francis, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, Petra Selmer, and Andrés Taylor. 2018. Cypher: An Evolving Query Language for Property Graphs. In *Proceedings of the 2018 International Conference on Management of Data* (Houston, TX, USA) *(SIGMOD '18)*. Association for Computing Machinery, New York, NY, USA, 1433–1445. https://doi.org/10.1145/3183713.3190657

[3] Wentao Han, Kaiwei Li, Shimin Chen, and Wenguang Chen. 2018. Auxo: a temporal graph management system. *Big Data Mining and Analytics* 2, 1 (2018), 58–71.

[4] Georgia Koloniari, Dimitris Souravlias, and Evaggelia Pitoura. 2013. On graph deltas for historical queries. *arXiv preprint arXiv:1302.5549* (2013).

[5] Maria Massri, Zoltan Miklos, Philippe Raipin, and Pierre Meye. 2021. Clock-G: A temporal graph management system with space-efficient storage technique. In *Proceedings of the 25th International Conference on Extending Database Technology. (To be submitted)*. Edinburgh, UK.

[6] Maria Massri, Zoltan Miklos, Philippe Raipin, and Pierre Meye. 2021. T-Cypher: A time-extended language for querying temporal property graph databases. In *Proceedings of the ACM SIGMOD/PODS International Conference on Management of Data. (To be submitted)*. Philadelphia, PA, USA.

[7] Luo Xiangyu, Luo Yingxiao, Gui Xiaolin, and Yu Zhenhua. 2020. An Efficient Snapshot Strategy for Dynamic Graph Storage Systems to Support Historical Queries. *IEEE Access* 8 (2020), 90838–90846.

Privacy over RDF datasets

Sara Taki, Cédric Eichler, Benjamin Nguyen
INSA Centre Val de Loire, LIFO
Bourges, France
first.last@insa-cvl.fr

KEYWORDS

Privacy, Anonymisation, Differential Privacy, Semantic Web, RDF, SPARQL, Graph projection

1 CONTEXT AND MOTIVATION

Nowadays, data is often organized as graphs with an underlying semantic to allow efficient querying, while supporting inference engines. Such is the case for linked data and the semantic web, which are built using the RDF standard [7]. An RDF data set is a set of triples (subject-predicate-object) which form a labeled directed graph. With the increased use of this representation, privacy in such data sources is becoming an issue [3]. A massive amount of work has focused on privacy in data presented as tables, resulting in multiple well-established models, such as k-anonymity [11], l-diversity [8], and differential privacy [4]. On the contrary, privacy in the context of semantic graph databases has been mildly studied.

In the context of this Ph.D., we study privacy over RDF datasets. In particular, we are interested in adapting differential privacy (DP) to edge-labeled directed graphs with an underlying semantic.

2 BACKGROUND

Differential Privacy. DP [4] is a model that provides a robust statistical definition of privacy tailored to the problem of privacy-preserving data analysis. The goal is to ensure that an attacker is not able to infer (beyond a certain probabilistic threshold) whether an individual contributed to the result of a query over a database. The exact protection and the notion of individuals' contributions are defined based on the concept of neighboring (or adjacent) databases.

Definition 2.1 (ϵ, δ-differential Privacy[4]). A randomized mechanism K: $D^n \rightarrow \mathbb{R}^d$ preserves (ϵ, δ)-differential privacy if for any pair of adjacent databases (x, y) $\in (D^n)^2$ and for all sets S of possible outputs:

Pr[K(x) $\in$ S] $\leq e^{\epsilon}$ Pr[K(y) $\in$ S] $+ \delta$ where the probability is taken over the randomness of K.

In traditional relational databases, two databases are neighbors if they differ by at most one individual. Hence, in that context, one should not be able to infer beyond a certain probability whether an individual is present in the database by observing the result of a DP-mechanism.

Adapting DP to graphs. Building DP algorithms on graph data means defining adjacency between graphs. The two most prevalent definitions proposed in the litterature are edge-DP and node-DP [5]. In node-DP,two graphs are neighbours if they differ by at most one node and all of its incident edges. In edge-DP, neighboring graphs are defined as graphs that differ by at most one edge. Previous studies [1, 6, 9, 12] show different approaches to work with DP in graphs. In [12], a new definition of adjacency is presented in the context of social graphs, called Out-link. In out-link-DP, neighboring graphs are defined as graphs that differ by the outlinks (outedges) of a chosen node. However, none of these approaches tackles edge-labelled graphs and their semantic. QL-out-edge-DP is a recent refinement of out-link-DP integrating semantic [10]. QL-out-edge-DP relies on the definition of the set QL of edge types (labels) that are considered sensitive. Two neighbouring graphs differ by the outedges whose labels are within QL of one node. Edges whose labels are not contained in QL are preserved.

Building graph-DP algorithms. There exists various way of making a mechanism DP. One particular general approach to achieve differential privacy is to add an appropriate amount of noise to the query results. This obviously depends on the query and the adjacency model. *Sensitivity* is the parameter that determines the magnitude of noise to be added. It represents how much the query result can change over neighbouring datasets. The prevalent approach is to add noise that depends on the *global sensitivity* (GS) of the query, which measures the maximum variation of the query result when evaluated upon *any* pair of neighboring databases. The main drawback with GS is that it can be very large, and sometimes unbounded (e.g. when computing the MAX degree w.r.t. node-DP). One technique to reduce the GS of a query is to first project the dataset from the initial space noted ginto a bounded space noted R, then run the query on the projected result. The sensitivity of the composition is bounded by the sensitivity of the projection times the sensitivity of the query on the projected space [6], where the sensitivity of the projection is the maximal distance between the projections of two neighboring graphs.

Theorem 2.2 (Sensitivity of the composed mechanism [6]). *Given a projection $T : g \rightarrow R \subseteq g$, a function $f : R \rightarrow \mathbb{R}^d$, and a distance d over g, noting $\Delta_d(f \circ T)$ the global sensitivity of $f \circ T$ w.r.t. d, we have:*

$$\Delta_d(f \circ T) \leq \Delta_d^R f \times \Delta_d T$$

Using this theorem, [2] have proposed to use projection by edge addition in order to achieve DP on undirected, unlabeled graphs.

3 OUR APPROACH : USING PROJECTION TO ACHIEVE DP FOR RDF GRAPHS

3.1 Objective of the work

The objective of this Ph.D. is thus to propose methods to query RDF graphs while respecting DP constraints. Our approach is to project RDF graphs into a bounded subspace in order to reduce

the sensitivity of the queries, and thus 1) make it possible to add a finite amount of noise when the sensitivity on the original space is infinite and 2) potentially improve the utility by reducing the amplitude of said noise. Our objective is to study several projection methods, propose theoretical bounds on noise to be added, and build a prototype implementing our system in order to check the utility of the results.

3.2 Current results

In our current work, we proposed and studied three edge-addition based graph projection methods named T_1, T_2 and T_3 which are appropriate w.r.t. node-DP, out-link-DP, and QL-out-link-DP, respectively. These projections bound the maximal degree (T_1), out-degree (T_2), and out-degree of specific labeled edges (T_3) of a input graph, the bound being noted D. Although the concept of projection by edge-addition is not new, since it was introduced by [2], previous work only considers undirected graphs with no labels on edges. On the contrary, we consider RDF graphs, i.e. edge-labeled graphs, and study the behaviour of our projections w.r.t. several privacy models.

Proposed projection algorithms. The algorithm 1 formalizes T_3 which is well adapted to RDF and serves as example. In general, our projections take as input a graph G and build a new graph with the same nodes as G but without any edges. They then try to insert each edge following an edge-ordering function (see below) A. An edge e from v_1 to v_2 labeled l is successfully inserted whenever its insertion preserves the constraint, i.e. (T_1) the degrees of both v_1 and v_2 are less than D; (T_2) the out-degree of v_1 is less than D; (T_3) l QL or v_1 has less than D out-edges with a label in QL.

Algorithm 1: T_3: projection by edge-addition, Bound QL-out-degree

Input: A graph G = (V, E) ∈ gwhose edges are label within the set L, a degree bound D, a stable edge ordering A, a set of labels QL ⊆ L

Output: An output D-Ql-out-degree bounded graph T_3(G)

$E^D \leftarrow \emptyset$; QLOutd(v)←0 for each v ∈V;
foreach *e=(v_1,l,v_2) ∈ A, in A's order* **do**
  if $l \in QL \wedge QlOutd(v_1) < D$ **then**
    $E^D \leftarrow E^D \cup${e};
    QLOutd(v_1)=QLOutd(v_1) 1;
  end if
  if l QL **then**
    $E^D \leftarrow E^D \cup${e};
  end if
end foreach
return G^D_{QL} = (V, E^D)

Projection sensitivity. We studied the sensitivity of each of the proposed projection in the context of node-, out-edge-, and QL-out-edge-DP. In particular, we demonstrated that the GS of T_2 in the out-edge-DP model and the GS of T_3 in the QL-ou-edge-DP model is 1. Using theorem 2.2, this shows that the GS of the composition is *at most* the GS of the query in the restricted space.

Edge ordering. Projection by edge-addition requires the definition of an order over the space of edges. This edge ordering must be stable in the sense that given two neighboring graphs G_1 and G_2, if two edges appear in G_1 and G_2 then their relative order must be the same in A(G_1) and A(G_2). Here, the order can be determined by combining three sub-orders on the sources (subjects), labels (predicates), and objects (destinations) which can simply be lexicographical.

Testing on queries. We have studied several RDF queries and have shown that unbounded GS can be bounded using our projection approach, but also that an original bounded GS can be very much reduced.

4 FUTURE WORK

In future work we mainly plan on implementing the system, performing large scale experiments, and investigating the utility of the resulting mechanisms in particular. More specifically, when the GS of the projection is 1, we plan on investigating the trade-off between the data-degradation due to the projection and the reduction of the GS of the query (and thus of the amplitude of the noise). Interestingly, even though edge ordering has no impact on the GS of the projection, we have found that it does impact utility. Another outlook is therefore to study this impact and propose optimal edge orderings.

5 ACKNOWLEDGMENT

Work supported by the French National Research Agency, under grant ANR-18-CE23-0010

REFERENCES

[1] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. 2013. Differentially private data analysis of social networks via restricted sensitivity. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*. 87–96.
[2] Wei-Yen Day, Ninghui Li, and Min Lyu. 2016. Publishing graph degree distribution with node differential privacy. In *Proceedings of the 2016 International Conference on Management of Data*. 123–138.
[3] Rémy Delanaux, Angela Bonifati, Marie-Christine Rousset, and Romuald Thion. 2018. Query-based linked data anonymization. In *International Semantic Web Conference*. Springer, 530–546.
[4] Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 4052)*, Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener (Eds.). Springer, 1–12. https://doi.org/10.1007/11787006_1
[5] Michael Hay, Chao Li, Gerome Miklau, and David Jensen. 2009. Accurate estimation of the degree distribution of private networks. In *2009 Ninth IEEE International Conference on Data Mining*. IEEE, 169–178.
[6] Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2013. Analyzing graphs with node differential privacy. In *Theory of Cryptography Conference*. Springer, 457–476.
[7] Graham Klyne and Jeremy J. Carroll. 2004. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/
[8] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. 2007. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1, 1 (2007), 3–es.
[9] Sofya Raskhodnikova and Adam Smith. 2015. Efficient lipschitz extensions for high-dimensional graph statistics and node private degree distributions. *arXiv preprint arXiv:1504.07912* (2015).
[10] Jenni Reuben. 2018. Towards a differential privacy theory for edge-labeled directed graphs. *SICHERHEIT 2018* (2018).
[11] Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 05 (2002), 557–570.
[12] Christine Task and Chris Clifton. 2012. A guide to differential privacy theory in social network analysis. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 411–417.

2

Facilitating Heterogeneous Dataset Understanding

Nelly Barret
Inria & Institut Polytechnique de Paris
nelly.barret@inria.fr

ABSTRACT

The era of Big Data and data sharing has lead to very large volumes of data becoming available to users across the world. This data is heterogeneous in its modelling, format and quality. Taking full advantage of such data raises many challenges, in particular related to the integration and the understanding of such data. My PhD thesis, started in January 2021, seeks to develop novel methods to help users without advanced IT skills discover a new dataset, by (*i*) building an abstract understanding of the data, as consisting of *records* and *collections*, (*ii*) interpreting or classifying the data based on users' interests, and leveraging Information Extraction and Natural Language tools.

1 INTRODUCTION

The Open Data Initiative has led to an increasing number of publicly-spread datasets. Such datasets are often quite large and heterogeneous (depending on the provider, the kind of data, etc). Many such datasets are large; further, they are extremely heterogeneous, in particular for what concerns their data model (RDF, JSON, XML, CSV, property graphs, relational databases, etc.), their schema (if a schema exists), etc. The scale and heterogeneity make it challenging for human users to identify, among the many available datasets, those that could be used for a given application they have in mind.

This thesis is part of the ConnectionLens project [1], which aims at integrating heterogeneous data into a graph. Our goal is to create small expressive descriptions of what a dataset is about, using the power of integration of ConnectionLens. In this paper, we present the challenges (Section 2), then the approach (Section 3) and finally some preliminary results (Section 4) before concluding (Section 5).

2 CHALLENGES

Finding the right dataset is complicated, especially because they are often not well-documented and it can be difficult to appreciate how it can be useful. Our approach, which aims at helping users to choose a dataset, should satisfy the following requirements:

- **R1: The approach should be applicable to any kind of data.** There are various data formats, such as RDF (as in the Open Data Cloud), but also XML (as in the PubMed database), JSON (most of French open data), relational databases, and so on. This requirement is handled by Section 3.1.
- **R2: The data descriptions we build for users should be sufficiently expressive, but also compact.** Users need to understand what is inside a dataset, but when a full description is complex, we need to bring them only the most important facts about it. We discuss how to fulfil this requirement in Sections 3.2 and 3.3.

3 APPROACH

ConnectionLens is a system capable to produce a graph G from any dataset of any format, where each node is a piece of data and edges link these nodes to reflect the content of the original source. Moreover, an entity extraction process is applied on text nodes, to extract from them named entities, such as `Person`, `Location`, `Organization`, `Date`, etc. In my thesis, to be able to produce compact descriptions of any data format, we leverage ConnectionLens to start our summarization method from the graph G. Our approach is the following:

(1) **Build a structural summary of G.** The structural summary G' is a graph computed out of G, potentially much smaller than G, and which gives us a first idea of groups of nodes that may contain similar information: each such group of G nodes is *represented* by a single node in G'.
(2) **Find collections and records.** Starting from the summary, we seek to identify the nodes that represent *records*, that is, objects of a certain "kind" with some internal structure, and *collections*, that is, containers of potentially many records of the same "kind".
(3) **Categorize collections.** Finally, we aim at *classifying collections* among a set of categories $\mathcal{K}$, containing (*i*) the kind(s) of data that the user is looking for, if the user can formulate such a request, e.g., "Books", or "Places to visit", and/or (*ii*) a set of generic categories we pre-define, such as `Person`, `Organization`, `Location`, `Event` and `Creative work`. The categorization adds a limited form of semantics (we keep things simple on purpose since we assume non-technical users), and enable adapting to the users' interest.

3.1 Summarization

We explain now how we compute the summary G' of G. For efficiency, we distinguish two cases: rooted, acyclic data source graphs, vs. the general case where graphs may have cycles and/or may not have a root.

Rooted acyclic graphs. These graphs are obtained for instance from XML or JSON datasets. On such graphs, we apply the strong DataGuide summarization method [4] to create G' from G. A Dataguide is a concise summary of the structure of a database. This method builds a set of paths, such that each path of the DAG appears exactly once in the summary. Such summarization method works only on acyclic graphs because the recursion should not encounter a cycle.

General graphs. Such graphs can originate in RDF, property graphs, or relational database datasets (where primary-foreign keys can lead to cyclic connections between the tuples). For such graphs, we need a graph summarization method that (*i*) reflects all the graph, (*ii*) groups nodes into equivalence classes and (*iii*) can be computed efficiently even from large graphs. RDFQuotient [3], originally introduced for RDF but easy to adapt to arbitrary graphs, meets these criteria, thus we rely on it to compute the summary G' of G for non-acyclic graphs. RDFQuotient gives a set of equivalence classes between nodes based on their types and their properties.

3.2 Records and Collections

We seek to understand G' based on two key concepts:

- A **Record** is basically a *thing*; in data modelling terms, it describes either an entity or a relationship. It has some properties (e.g. a `title` and a `DOI` for a paper) and can handle nested collections (e.g. the authors list of a paper).
- A **Collection** is a set of similar records (e.g. a bibliography is a collection of books). They are *explicit* when a node handles similar records; or *implicit* when some records refer to the same purpose without being handled by a node.

Other nodes in G' are called Sub-Records and are mainly the properties of the records (i.e. the set of outgoing properties of a record r, referred as $r.\mathcal{P}$). Furthermore, we compute the *signature* of each sub-record s, where the signature is compound of a *domain* ("*to which categories s belongs to?*") and a *range* ("*to which categories s points to?*"). For example, the sub-record `settledDownIn` has for domain {`Person, Organization`} and for range {`Location`}.

To find them, we first determine collections and then, in a top-down fashion, the direct children of collections are identified as records. To compute collections, we rely on a clustering algorithm we devised, based on the *support* of a set of properties among a set of potential records (how many of these records have this set of properties). Our clustering algorithm identifies both *explicit* collections, where a G' node is actually the parent of all the nodes representing the records in the collection, and *implicit* collections, where such a common parent/collection node does not exist in G'.

3.3 Analysis and Categorization of Collections

Given a set of hints $\mathcal{H}$ and a set of user-defined categories $\mathcal{K}$, we aim at categorizing a collection c among $\mathcal{K}$, i.e. give a category $k \in \mathcal{K}$ to c using $\mathcal{H}$, as illustrated by Algorithm 1. A **hint** h is a triple $\langle A, l, B\rangle$ where A is the *domain* $\subseteq \mathcal{K}$, l is the label and B is the *range* $\subseteq \mathcal{K}$. For instance, the hint ⟨Organization, hasCEO, Person⟩ states that a collection having a record holding the property `hasCEO`, whose signature's range matches `Person`, should be categorized as an `Organization`.

For each record $r \in c$, we initialize $\mathcal{K}_r$ (set of candidate categories in which r may belong) and scores (score of *nc* for each hint in $\mathcal{H}$). Then, if r has a label semantically close to one of the category in $\mathcal{K}$, this category is stored as a candidate category in $\mathcal{K}_r$. For each child $nc \in r$, we create a pair π containing the label and the signature of *nc*. Then, we compute the similarity of π with each hint h in $\mathcal{H}$, where the similarity is based on the label and the signature of both elements. We choose the hint h leading to the highest similarity score for each π. Each category indicated by the domain of h gets a vote. Then, we classify r in the category that gets the highest number of votes or `Other` if no category is frequent enough. Finally, we classify c in the most represented category in its records. We also determine if a collection describes entities or relationships, by looking at the connections between the collections.

4 STATUS

We have fully implemented our approach in a prototype, which leverages the graph creation and storage of ConnectionLens [1], and includes the novel algorithms described in Section 3. More details can be found in a short paper [2].

Figure 1 shows an example of our approach applied on a set of PubMed articles. The set of articles is considered as a collection of

Algorithm 1: Classifying a collection c

Input: a collection c, hints $\mathcal{H}$, categories $\mathcal{K}$
Output: a category $k \in \mathcal{K}$ or `Other`

foreach $r \in C$ **do**
  $\mathcal{K}_r \leftarrow \emptyset$
  scores $\leftarrow \emptyset$
  foreach $k \in \mathcal{K}$ **do**
    if *the similarity between k and the label of r is higher than a threshold* **then**
      $\mathcal{K}_r \leftarrow \mathcal{K}_r \cup \{k\}$
  foreach $nc \in r.children$ **do**
    $\pi \leftarrow (nc.label, nc.signature)$
    foreach $h \in \mathcal{H}$ **do**
      scores $\leftarrow$ scores $\bigcup(h, sim(\pi, h))$
    bestHint $\leftarrow$ argmax(scores)
    $\mathcal{K}_r \leftarrow \mathcal{K}_r \cup \{$ bestHint.domain $\}$
  Classify r in the most frequent $k \in \mathcal{K}_r$, or `Other`
Classify c with the most frequent category of its records

`Creative Work`. Moreover, the authors are identified as a collection of `Persons`.

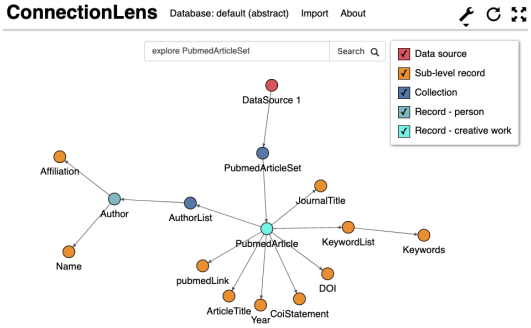


Figure 1: Example of G', an abstract graph with collections and categorized records.

5 CONCLUSION AND PERSPECTIVES

My PhD thesis aims to create expressive descriptions of big heterogeneous datasets by using summarization methods and categorization of expressive structures (records and collections). Beyond finalizing the implementation of our platform for all the data models we consider, we will experiment to analyse its scalability as well as the expressiveness and precision of the record categorization. Next, we will investigate the adoption of sampling-based approaches, to try to construct such dataset descriptions without traversing the dataset entirely to further improve performance.

Thesis context My PhD is funded by DIM RFSI and is a collaboration between Inria and WeDoData, a SME specialized in data visualization and interactive data-driven Web content. My PhD advisers are Ioana Manolescu (Inria) and Karen Bastien (WeDoData).

Acknowledgments. This work is funded by DIM RFSI PHD 2020-01 and AI Chair SourcesSay project (ANR-20-CHIA-0015-01) grants.

REFERENCES

[1] A. C. Anadiotis, O. Balalau, C. Conceicao, H. Galhardas, M. Y. Haddad, I. Manolescu, T. Merabti, and J. You. Graph integration of structured, semistructured and unstructured data for data journalism. *Information Systems*, page 42, July 2021.
[2] N. Barret, I. Manolescu, and P. Upadhyay. Toward generic abstractions for data of any model. 2021. Short paper, submitted for publication at BDA 2021.
[3] F. Goasdoué, P. Guzewicz, and I. Manolescu. RDF graph summarization for first-sight structure discovery. *The VLDB Journal*, 29(5), Apr. 2020.
[4] R. Goldman and J. Widom. Dataguides: Enabling query formulation and optimization in semistructured databases. In *VLDB*, 1997.

Towards a Logistical View for Data Lake Optimization

Marzieh Derakhshannia*
Anne Laurent*
Marzieh.derakhshannia@etu.umontpellier.fr
Anne.laurent@umontpellier.fr
LIRMM, Univ Montpellier, CNRS
Montpellier, France

ABSTRACT

Data life cycle management is a challenging issue in the big data arena. Data lakes are the new generation data repositories that provide the platform to respond to the management of huge amounts of raw data. In terms of structure, data lake architecture could have a significant role in improving the data quality and service levels. For this reason, suitable data lake design and appropriate management strategies play the important role in implementing of fruitful data lake. Logistical view is a proper solution to manage an optimal data lake that will be favorable for organisations.

CCS CONCEPTS

• **Information systems** → **Storage architectures**; **Data management systems**.

KEYWORDS

Data Lake, Data Lake Architecture, Logistic Systems, Supply Chain, Supply Chain Management

1 CONTEXT

In the last decades, the data has become a valuable property for organizations and is considered as digital oil in the big data environment. To make capital from this precious product, proper equipment is required [6]. Traditional data management tools have inadequacies in gathering, storing, processing, and visualizing all types of data in an integrated manner in the big data arena. Therefore, the term " Data lake " has been emerged to address the shortcomings flaws and meet all the challenges related to the data lifecycle [10].

Data lakes are the developed, affordable, and agile generation of centralized data management systems that provide plenty of faculties to conduct the procedures of storage and knowledge mining from a huge amount of heterogeneous data [4, 9, 12]. However, welcoming all kinds of data in their native formats increases the risk of saturation with unmanageable or useless data; a phenomenon is known as “data swamp”; and threatens the security of sensitive data. For this reason, data management strategies and architecture design are more highlighted in data lake productivity. The architecture of the data lake and the administration regulations could have significant impacts on the efficiency of data lake performance. Thus, an optimized architecture under proper managerial disciplines enhances the data lake credibility and reinforces its reliability as a new component of the information systems to support decision-making processes in the organizations.

2 CHALLENGES

Data lake architecture design and data management are problematic issues for the implementation of an efficient data lake infrastructure. Based on these requirements, many studies in the field of the data lake architecture design have been conducted in parallel with the technical implementation [5, 7–9, 11, 12]. Despite many types of researches on the data lake architecture and with regard to their cons and pros, optimization of the data lake structure and performance based on managerial and mathematical viewpoints has been less considered. For this reason, we suggest logistical approaches like product lifecycle management and hybrid decision-making strategies (like facility location-allocation problems) based on supply chain management principles to design and optimize the data lake architecture. The significant principle of supply chain management is to minimize the total costs related to the flow of products in such a way that profitability of the chain and quality of the service requirement, will be maximized. We intend to implement this principle for data lake management due to the new proposed architecture and imitated strategies.

3 METHODS

Since the data could be considered as a valuable product, we can imagine that a data lake performs like a logistic system that manages the data throughout its life cycle. This viewpoint brings about a great deal of systematic and managerial strategies to develop data lake architecture and gain the integrated manner to manage and govern the data in a cost-effective platform. Based on this requirement, the supply chain is one of the well-known logistic systems that present a systematic view of logistic parties who are integrated into a unified network to produce a commodity or provide a service [1, 2]. To design the profitable supply chain network, each level of this chain is managed by tactical and strategic strategies. Thus, the supply chain management strategies prepare the suitable guidelines to design and manage the data lake architecture in general or in detail scales.

The processes of designing a data lake architecture are related to positioning the different components like data ingestion, data storage, data process, and data access stages and defining the proper strategies and protocols to govern the data like metadata management or data democracy. For this reason, the data lake platform based on systematic definitions and network design concepts could

*Both authors contributed equally to this research.

be an effective step in the evolution of data lake architecture. Regarding the analogy of data lake and supply chain, we consider the data like a product or service that should be monitored through several stages in the data lake architecture from data entering until data consumption. With respect to this point of view, we can benefit from some management tools to improve data lake functionality. For example, we can use the policies of product lifecycle assessment that is employed in logistic systems, for data governance in data lake or hybrid decision-making problems that are essential in supply chain network design, for implementation and optimization of the data lake platform.

As figure 1 shows the simple analogy of supply chain structure and data lake architecture, data are supplied, stored, manipulated, monitored, and delivered in data lake as a product that is managed throughout the supply chain. In this architecture, data are produced and supplied from different sources and are entered in the data lake by the ingestion stage, then they are stored in their raw formats at the storage level, and finally, they are sent to the processing stage for further procedures based on users requirements. In this architecture, each level of data life cycle could be managed by mimetic logistical patterns. For example, the ingestion strategies are related to the optimal decisions about metadata management and data modeling, like supply and product design management in the supply chain. The storage strategies concentrate on selecting cost-effective storage strategies (single-store systems or multi-store systems) and data integration, just like decision-making about the number, the location, and the type of the warehouses and inventory management in the supply chain. The process strategies are the vital decisions that impact on quality of data and veracity of extracted knowledge. Therefore, the main goal of process strategies is related to choose the data processing methods that prepare the data with the fastest time and the least cost possible for data queries. Based on this requirement, the hybrid decision-making models like location-allocation problems; that bring about the best solution to optimize the supply chain networks [3]; could be a logical solution to design the affordable data lake architecture. For instance, we can use the techniques of supply chain network design (like optimization mathematical models) to determine the structure of data lake with a variety of strategic decisions such as determining the number of proper processing jobs, tactical decisions like data governance all over the data lake just like product quality assessment in the supply chain, and operational decisions like fulfilling user demands. The main objective of designing the logistical data lake architecture is to minimize further costs related to the data ingestion, storage, and processing and maximize the level of data lake performance.

4 CONCLUSION

In the direction of mentioned goals, we focus on data lake architecture design due to the mathematical optimization models that are inspired by hybrid problems in supply chain network design. The accomplished studies showed that these interdisciplinary methods smooth the path of presenting data lake optimization models and guarantee to obtain the affordable data lake architecture as a logistic system.

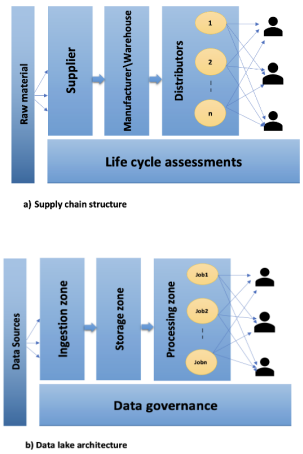


Figure 1: Analogy of supply chain structure and data lake architecture

REFERENCES

[1] Benita M Beamon. 1998. Supply chain design and analysis:: Models and methods. *International Journal of Production Economics* 55, 3 (1998), 281 – 294. https://doi.org/10.1016/S0925-5273(98)00079-6

[2] Sunil Chopra and Peter Meindl. 2007. Supply chain management. Strategy, planning & operation. In *Das summa summarum des management*. Springer, Germany, Springer Gabler, 265–275.

[3] Mark S Daskin. 2011. *Network and discrete location: models, algorithms, and applications*. John Wiley & Sons, 111 River St, 07030 Hoboken.

[4] Huang Ling Fang. 2015. Managing data lakes in big data era: What's a data lake and why has it became popular in data management ecosystem. *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)* „ 15503297 (2015), 820–824.

[5] Corinna Giebler, Christoph Gröger, Eva Hoos, Rebecca Eichler, Holger Schwarz, and Bernhard Mitschang. 2021. The Data Lake Architecture Framework: A Foundation for Building a Comprehensive Data Lake Architecture. In . Conference for Database Systems for Business, Technology and Web (BTW), 70469 Stuttgart, Germany. https://doi.org/10.18420/btw2021-19

[6] Dennis D Hirsch. 2013. The glass house effect: Big Data, the new oil, and the power of analogy. *Me. L. Rev.* 66 (2013), 373.

[7] Bill Inmon. 2016. *Data Lake Architecture: Designing the Data Lake and avoiding the garbage dump*. Technics publications, .

[8] Jabrane Kachaoui and Abdessamad Belangour. 2020. From single architectural design to a reference conceptual meta-model:an intelligent data lake for new data insights. *International Journal* 8, 4 (2020), 1460–1465.

[9] Alice LaPlante and Ben Sharma. 2016. *Architecting data lakes: data management architectures for advanced business use cases*. O'Reilly Media, 1005 Gravenstein Highway North, 95472 Sebastopol.

[10] Pasupuleti Pradeep. 2015. *Data lake development with big data : explore architectural approaches to building Data Lakes that ingest, index, manage, and analyze massive amounts of data using Big Data technologies*. Packt Publishing, Birmingham.

[11] Pegdwendé Sawadogo and Jérôme Darmont. 2021. On data lake architectures and metadata management. *Journal of Intelligent Information Systems* 56, 1 (2021), 97–120.

[12] John Tomcy. 2017. *Data Lake for enterprises*. Packt Publishing, Birmingham.

Why-Not explanations for recommenders

Hervé-Madelein Attolou
herve-madelein.attolou@cyu.fr
ETIS, CY Cergy Paris University, ENSEA, CNRS UMR8051
Cergy-Pontoise, France
Supervisors: Dimitris Kotzinos & Katerina Tzompanaki

ABSTRACT

Recommenders suggest pertinent items to users from a vast variety of possibilities. However, it is crucial for the user and the system developer to understand why the system recommends certain items (*why*), and why it does not recommend others that he/she might expect (*why not*). In this thesis, we aim to explore explanations to the *why not* problem, which is less explored, but equally important.

KEYWORDS

Explanations; Why-Not questions; Explainable AI; Recommenders

1 INTRODUCTION

Recommenders are automatic tools that aid users in their data exploration tasks, usually in scenarios where the possibilities are vast, e.g., on e-commerce and other online platforms. Recommenders provide suggestions for data items based on explicit or implicit user feedback, like their preferences expressed as likes or ratings, previous views and purchases, their social network, or other context variables, e.g., spatiotemporal features, user characteristics, e.t.c.

Unexpected (either existing or missing) and *not-explained* recommendations may frustrate the user and can be detrimental to his/her trust and loyalty to the system. Explaining existing recommendations (and in general machine learning results) is an active scientific problem for at least the last decade (see Section 2). In this thesis, we focus on the problem of explaining *missing recommendations*, which we argue can be as important as explaining recommendations.

In our context a missing recommendation would be any item that the user or the system designer expect to be part of the recommendation and is not. It could be missing for "good" reasons, which would be justified by the provided explanation.

Consider for instance an e-recruitment site, which does *not* propose managerial positions to Alice. For Alice to trust the system, and continue using it, she needs to understand why this kind of positions is not proposed. Are there some points in her CV that count a lot for the system and thus need improvement? Or is the system gender biased?

Furthermore, explanations of missing recommendations can reveal why certain (categories of) items are neglected by the system. For example, consider a movie streaming service that suggests only romantic movies to Alice (Figure 1). Explaining why no action movies are suggested, can be a useful tool for the system designers, who may want to debug their system or improve its diversity.

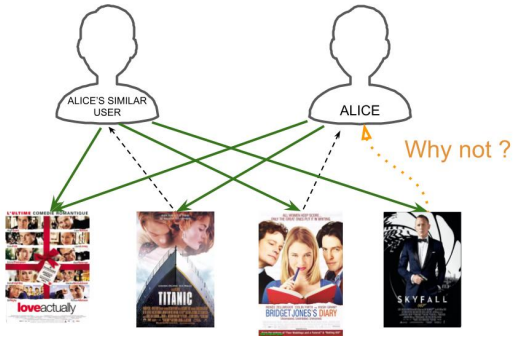


Figure 1: User likes (solid arrow), recommendations (dashed arrow) and missing recommendation (dotted arrow)

In both examples, the explanations benefit the user and/or the system developer. In this thesis, we aim to explore explanations for missing recommendations from the user and the system designer viewpoint, for various types of recommenders. Typically the search space of explanations on missing items is big, thus we aim to design efficient algorithms to compute the most pertinent explanations.

2 RELATED WORK

Recommenders. Content-based recommenders [5] exploit similarities among the characteristics of the item and user profiles. Score-based (or collaborative filtering) recommenders [8] exploit ratings and other quantifiable user-system interactions given to the items by the users. Context-based recommenders exploit information related to the situation that the items and/or the users are in, their spatio-temporal configuration, etc. [11].

Explainable Recommenders. So far, the research in explainable recommendations is focused on the Why question: "Why is an item recommended?". The approach to explain a system depends on whether the system is *white*, i.e., exposing the information about its internals, or *black*, i.e., only the input and the output are known [12]. In the case of black-box recommenders, the explanations consist in interpreting the results by revealing relationships in the input data [4], or by pinpointing the importance of the different features concerning the predicted value [6]. Another line of work proposes to locally explain a system, with a simpler machine learning model [7]. In the case of a white box system, the explanations consist in dwelling into the intrinsic characteristics of the recommendation system to truly explain the system [2].

Why-Not questions. Explaining and debugging queries in the absence of expected query results has been studied in traditional databases [3, 10], top-k spatial queries [4], or workflow analysis [1].

Recently, [9] has introduced Why-Not questions in recommenders, providing a method to answer them in Nearest Neighbours collaborative filtering. Still, a lot of space is left to be explored.

3 RESEARCH PROBLEM

A *Why-Not* question is either a question of set containment, both for individual set items and subsets, or a question of ranking (why something is ranked above or below something else)

For example, a user may ask Why-Not one missing item, a category of missing items, or ask Why-Not a certain item (or category) instead of another one already appearing in the recommendations. Additionally, he/she may ask why a *recommended* item does *not* appear higher in the recommendation list, or higher than another recommended item [9]. From another aspect, a Why-Not question can be defined in the scope of one user or for the whole system, depending also on who is the user asking the question, i.e., the final user of the recommendation system or the system developer. In this thesis, we are going to investigate the various views of a Why-Not question.

An explanation is generally defined as a piece of information displayed to users, providing the reasons why a particular item is recommended [12]. In the same spirit, a *Why-Not explanation* is information that helps users understand the reasons behind the absence of interesting items.

Consider again the example of the e-recruitment website in Section 1. An explanation of "Why do I get recommended to apply to an entry-level job?" could be a sample of the input data (line of the CV or extract of a job offer), or a sample of the training data (job offer to a user with a similar curriculum). A similar question, but with a different scope, would be to ask "Why don't I get any senior level job offers?". A possible explanation to this question could coincide with the aforementioned explanations for the why question. However, the why explanations do not exactly answer the Why-Not question; the user expects an answer in connection to what he/she is missing. A better type of explanation could be to exploit explicit or implicit negative user feedback for the items or categories in question, for instance "You have rarely viewed a senior-level position offer". Alternatively, to provide the users with more concrete, actionable information, the explanations could take the form of actions that if were performed by the user, then the expected items would appear as recommendations. For instance, "If you had followed more management courses, you would have been proposed 'Senior Data Architect @B' position instead of 'Junior Java Developer @A'".

The challenges that exist in explaining a recommendation system via Why-Not questions are the following. First, as in any big-data system, it is hard to know what could be missing from a result, so as to formulate a Why-Not question. Second, the search space of an explanation for a missing recommendation is much bigger than for explanations for existing recommendation (multiple possible lineages of non-results). Third, there exist various recommendation system models, calling for different solutions for each model when considered as white boxes.

An approach towards addressing the first challenge could be an interactive framework that proposes candidate missing items to the users, employing machine learning for prediction or more traditional item-scoring methods. Clustering and pruning methods of the possible explanations can be used to address the second challenge. For the third challenge, we should study explainable models with different levels of detail, from fine-grained to coarse-grained, depending on the final consumer of the explanation. Such explanations could combine relevant points of the (training) data and their features, hyper-parameters of the algorithms, the model complexity, etc. In the same context, visualization and analysis of explanations must be considered for easier consumption and understanding of explanations by the final receiver.

4 OUTLOOK

Big data and artificial intelligence comes with the promise to improve people's lives, also by enhancing the discovery of interesting information and providing results tailored to users' profiles. However, the same technology, if not used responsibly, may lead to discrimination, amplify biases in the original data, restrict decision transparency and strengthen unfairness. With this project, we aim to promote research towards explainable, transparent, and fair systems by explaining the absence of certain options from a user's recommendation list.

REFERENCES

[1] Adriane Chapman and H. V. Jagadish. 2009. Why Not?. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data* (Providence, Rhode Island, USA) *(SIGMOD '09)*. Association for Computing Machinery, New York, NY, USA, 523–534. https://doi.org/10.1145/1559845.1559901

[2] Azin Ghazimatin, Oana Balalau, Rishiraj Saha Roy, and Gerhard Weikum. 2020. PRINCE: Provider-side Interpretability with Counterfactual Explanations in Recommender Systems. *Proceedings of the 13th International Conference on Web Search and Data Mining* (Jan. 2020), 196–204. https://doi.org/10.1145/3336191.3371824 arXiv: 1911.08378.

[3] Md Islam. 2013. On answering why and why-not questions in databases. *Proceedings - International Conference on Data Engineering*, 298–301. https://doi.org/10.1109/ICDEW.2013.6547468

[4] Yanhong Li, Wang Zhang, Changyin Luo, Xiaokun Du, and Jianjun Li. 2021. Answering why-not questions on top-k augmented spatial keyword queries. *Knowledge-Based Systems* 223 (2021), 107047. https://doi.org/10.1016/j.knosys.2021.107047

[5] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2011. *Content-based Recommender Systems: State of the Art and Trends*. Springer US, Boston, MA, 73–105. https://doi.org/10.1007/978-0-387-85820-3_3

[6] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 4768–4777.

[7] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (Eds.). ACM, 1135–1144. https://doi.org/10.1145/2939672.2939778

[8] J. Ben Schafer, Joseph Konstan, and John Riedl. 1999. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce (EC '99)*. Association for Computing Machinery, New York, NY, USA, 158–166. https://doi.org/10.1145/336992.337035

[9] Maria Stratigi, Aikaterini Tzompanaki, and Kostas Stefanidis. 2020. Why-Not Questions & Explanations for Collaborative Filtering. In *WISE*. Amsterdam, Netherlands. https://doi.org/10.1007/978-3-030-62008-0_21

[10] Aikaterini Tzompanaki. 2015. *Réponses manquantes : Débogage et Réparation de requêtes. (Query Debugging and Fixing to Recover Missing Query Results)*. Ph.D. Dissertation. University of Paris-Saclay, France.

[11] Minghua Xu and Shenghao Liu. 2019. Semantic-Enhanced and Context-Aware Hybrid Collaborative Filtering for Event Recommendation in Event-Based Social Networks. *IEEE Access* 7 (2019), 17493–17502. https://doi.org/10.1109/ACCESS.2019.2895824

[12] Yongfeng Zhang and Xu Chen. 2020. Explainable Recommendation: A Survey and New Perspectives. *INR* 14, 1 (March 2020), 1–101. https://doi.org/10.1561/1500000066 Number: 1 Publisher: Now Publishers, Inc.

Example Generation for JSON Schema

Lyes Attouche*†
Université Paris-Dauphine, PSL Research University
lyes.attouche@dauphine.fr

ABSTRACT

JSON is one of the most popular format used for exchanging data on the web, its schema-less representation of data is one of its advantages but also one of its drawbacks, since it can induce the creation of malformed data, but with the creation of JSON Schema it has become possible to describe and control the content of the JSON documents, which ensures the quality of the exchanged data.

JSON Schema is a logical language, and as any other logical language it is subject to some challenging problems like schema consistency and schema equivalence. A tool for generating documents from a JSON Schema can be useful when answering these theoretical problems, it also has a practical interest since it can serve for testing the behavior of JSON-based web APIs.

We present in this paper an approach for achieving documents generation from JSON Schema and provide some potential use cases.

KEYWORDS

JSON, JSON Schema, Data Generation

1 INTRODUCTION

In recent years many web APIs have adopted JSON as a data format for transmitting and receiving data to and from web servers.

The rise of its popularity is mainly due to its adoption of a schema-less representation of the data which ensures several advantages, but these advantages go along with many disadvantages, indeed the absence of a schema imply not having a control over the data, hence web APIs may receive corrupt and malformed data.

The growing popularity of JSON and the need for overcoming some of its disadvantages has led to the creation of JSON Schema [4]. As for any other data format endowed with a schema language, JSON Schema offers a way for interpreting and validating JSON documents. On the one hand JSON Schema is not widely adopted, one reason for this is that JSON Schema is considered to be hard to work with, on the other hand new web APIs tend to use it since it can improve their designs by offering means for defining validation rules and for contract testing.

Having a JSON data generator can serve for simulating the behavior of programs, either for performance measurements or for detecting errors that might rarely occur in a real world environment. Other than verification and testing purposes, another motive of this generator would be the generation of synthetic data that mimics real world data [8].

2 PRELIMINARIES

2.1 JSON data model

JSON values are either basic values, objects, or arrays. Basic values include the null value, booleans, numbers and strings. Objects represent sets of members, each member being a name-value pair, and arrays are ordered lists of values.

Example 1. *Consider the following JSON object*

```
{
    "x" : 0,
    "y" : true
}
```

This JSON object has two key-value pairs, where the key "x" is of type *number*, and the key "y" is of type *boolean*.

2.2 JSON Schema

JSON Schema is a language for defining the structure of JSON documents. It is maintained by the Internet Engineering Task Force IETF [3]. Its latest version has been produced on 2019-09 [9].

Example 2. *Consider the following schema*

```
{
  "type":"object",
  "properties": {"x": { "type": "number" }},
  "minProperties": 1
  "required": ["x"]
}
```

JSON Schema uses the JSON syntax. A schema is a JSON object whose fields are assertions keywords, here we have four of them at the outermost level: *type*, *properties*, *m*inProperties and *required*. The keyword *type* specifies the data type for a schema (e.g. the property "x" is of type *number*). Considering the other assertions cited previously, they are specific to the type *object* (i.e. can only be applied to a value of type *object*), in the example above *minProperties* constrains the JSON value to have at least one property. Such constraints exist for the other types (e.g. *minLength* and *maxLength* are used to constrain the length of a *string* value, *minimum* and *maximum* can only be applied to numeric values, etc.).

JSON Schema allows combining assertions using standard boolean connectives: *not* for negation, *allOf* for conjunction, *anyOf* for disjunction, and *oneOf* for exclusive disjunction. Moreover, indicating the set of accepted values can be done using the *enum* constraint.

*Mohamed-Amine Baazizi, Sorbonne Université, LIP6 UMR 7606, baazizi@ia.lip6.fr
†Dario Colazzo, Université Paris-Dauphine, PSL Research University, dario.colazzo@dauphine.fr

3 EXAMPLE GENERATION

Example generation for JSON Schema is the procedure of generating valid JSON documents w.r.t a JSON schema given as an input.

This work relies on some prior works on not-elimination and witness generation for JSON Schema [6], which respectively deal with the problem of removing the negation operators from a JSON Schema and generating a valid document w.r.t a JSON Schema (when the schema is non-empty).

More in detail, the structure of the witness generation algorithm is as follow : first the input Schema is translated to an algebraic representation that makes not-elimination possible. Afterwards, the algebraic representation undergoes a series of transformations until it reaches the last step of the process which is the generation step, which input is an environment composed of a set of variables along with their definitions. Finally, we recursively generate a JSON value for each variable, and these values will compose the witness of the schema.

Approach. The two problems of witness and examples generation have clearly the same nature, thus the algorithms and procedures for achieving example generation will rely on the ones introduced in witness generation, and the experimental analysis of the witness generation tool can help instigating either optimizations or new approaches.

Given the eventual existence of schemas with infinite number of valid instances, and since this set of instances is countably infinite the example generation process may be parametric by relying on user-defined constraints in order to retrieve a finite number of JSON documents.

The approach for generating values for the *null* and *boolean* types is trivial given that they have a finite number of possible values. For the *number type*, the idea is to find a valid value w.r.t the corresponding requirements to start with, then increment/decrement with a step which value gives the next/previous valid value, a user-defined constraint here could be a probabilistic distribution (e.g. Poisson distribution). Regarding the generation of *string type* values, the technique would be to update the *pattern* given in the constraints by excluding the word generated at each iteration until the desired number of JSON values is reached. The approach for generating multiple JSON values for the JSON types *array* and *object* is not yet well defined, but as stated before it will rely on the witness generation works with the possibility of introducing optimizations and new ideas.

4 USE CASES

4.1 Black box testing

Black box testing of an API is testing this API with no prior knowledge of its internal workings, it only focuses on its input and output.

Adopting JSON Schema as a way for representing the structure of the data in the JSON-based web APIs may not be a difficult task, given that these APIs provide data that is encoded using JSON.

Let us consider the two JSON Schemas that represent the input and output data consumed and produced by a web API, and then we proceed in streaming as follow : we generate data from the input JSON Schema and then we validate the data produced against the output schema. With this process we can easily check the behavior of the program, and also conduct performance measurements.

4.2 Synthetic data generation

Designing robust and reliable machine learning models require a lot of data, however acquiring huge amount of data is sometimes challenging, either for privacy and confidentiality concerns or simply for the scarcity of real world datasets.

To address this problem, companies and researchers tend to use synthetic data. Synthetic data is data that is artificially created, it can be fully synthetic or generated according to real world data, thus having the same structure and statistical properties [7].

Generating JSON synthetic data by applying similar techniques could be interesting, an approach here is inferring a JSON Schema from JSON data as well as structural and quantitative information [5], then we generate the synthetic data from the JSON Schema inferred by considering the information retrieved as constraints.

5 RELATED WORK

There exist few implementations of JSON data generators, and only a couple of them generate data from JSON Schema [1, 2]. These generators do not possess a solid theoretical background, hence they are not complete and they sometimes produce wrong outputs. The inability of eliminating negative operators is one of the struggles encountered by these generators, indeed when testing schemas containing negation the generated JSON documents are most of the time incorrect.

6 CONCLUSION

We have defined in this paper the problem of generating JSON data from JSON Schema and given an approach to achieve this. We have shown that although JSON Schema is not widely used for describing and validating JSON documents, there still exist use cases that could benefit from a JSON data generator. Generating mock data could be used for testing, by simulating the behavior of JSON-based web APIs in order to detect errors that could not be detected using real world data and to measure the performance of the programs. In addition to this, the generation of JSON data in the fields of artificial intelligence and machine learning could help overcome the problem of real world data scarcity by building synthetic datasets, and could also help in building solid AI automation tools.

REFERENCES

[1] Json data generator. https://github.com/json-schema-faker/json-schema-faker/.
[2] Json data generator. https://github.com/jimblackler/jsongenerator.
[3] Internet engineering task force, 2020. Available at https://www.ietf.org.
[4] Json schema, 2020. Available at https://json-schema.org.
[5] M. Baazizi, Dario Colazzo, G. Ghelli, and C. Sartiani. Counting types for massive json datasets. *Proceedings of The 16th International Symposium on Database Programming Languages*, 2017.
[6] Mohamed-Amine Baazizi, Dario Colazzo, Giorgio Ghelli, Carlo Sartiani, and Stefanie Scherzinger. Not elimination and witness generation for json schema, 2021.
[7] Fida Dankar and Mahmoud Ibrahim. Fake it till you make it: Guidelines for effective synthetic data generation. *Applied Sciences*, 11:2158, 02 2021.
[8] Srđan Popić, Ivan Velikic, Nikola Teslic, and Bogdan Pavkovic. Data generators: a short survey of techniques and use cases with focus on testing. 08 2019.
[9] A. Wright, H. Andrews, and B. Hutton. JSON Schema validation: A vocabulary for structural validation of json - draft-handrews-json-schema-validation-02. Technical report, Internet Engineering Task Force, sep 2019.

8 Prix BDA 2021

8.1 Prix des articles de recherche

BDA a la particularité de proposer deux catégories d'articles : les articles originaux non publiés et les articles publiés récemment dans une conférence internationale de renom. Cette dernière catégorie permet de diffuser largement les travaux faisant la renommée internationale de notre communauté nationale en gestion de données.

Lauréats du prix des articles de recherche

Article non publié : On Predictive Explanation of Data Anomalies, *Nikolaos Myrtakis, Ioannis Tsamardinos and Vassilis Christophides*

Article non publié : Significance and Coverage in Statistically-Sound Group Testing, *Nassim Bouarour, Idir Benouaret and Sihem Amer-Yahia*

Article publié à KDD'21 : Efficient Incremental Computation of Aggregations over Sliding Windows, *Chao Zhang, Reza Akbarinia and Farouk Toumani*

Article publié à SIGMOD'21 : HADAD : A Lightweight Approach for Optimizing Hybrid Complex Analytics Queries, *Rana Al-Otaibi, Bogdan Cautis, Alin Deutsch and Ioana Manolescu*

8.2 Prix des démonstrations

Lauréat du prix des démonstrations

ADESIT : Visualize the Limits of your Data in a Machine Learning Process, *Pierre Faure-Giovagnoli, Marie Le Guilly, Vasile-Marian Scuturici and Jean-Marc Petit*

8.3 Prix des thèses en gestion de données

Lauréats du prix des thèses

Prix de thèse : Rituraj Singh pour sa thèse intitulée
« *Data-centric workflows for crowdsourcing applications* ».
encadrants : Loïc Hélouët et Zoltan Miklos