SIGMOD Officers, Committees, and Awardees

Chair

Yannis Ioannidis
University of Athens
Department of Informatics
Panepistimioupolis, Informatics Bldg
157 84 Ilissia, Athens
HELLAS
+30 210 727 5224
<yannis AT di.uoa.gr>

Vice-Chair

Christian S. Jensen
Department of Computer Science
Aarhus University
Åbogade 34
DK-8200 Århus N
DENMARK
+45 99 40 89 00
<csj AT cs.aau.dk >

Secretary/Treasurer

Alexandros Labrinidis
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260-9161
PA 15260-9161
USA
+1 412 624 8843
<labrinid AT cs.pitt.edu>

SIGMOD Executive Committee:

Sihem Amer-Yahia, Curtis Dyreson, Christian S. Jensen, Yannis Ioannidis, Alexandros Labrinidis, Maurizio Lenzerini, Ioana Manolescu, Lisa Singh, Raghu Ramakrishnan, and Jeffrey Xu Yu.

Advisory Board:

Raghu Ramakrishnan (Chair), Yahoo! Research, <First8CharsOfLastName AT yahoo-inc.com>,
Amr El Abbadi, Serge Abiteboul, Rakesh Agrawal, Anastasia Ailamaki, Ricardo Baeza-Yates, Phil Bernstein, Elisa Bertino, Mike Carey, Surajit Chaudhuri, Christos Faloutsos, Alon Halevy, Joe Hellerstein, Masaru Kitsuregawa, Donald Kossmann, Renée Miller, C. Mohan, Beng-Chin Ooi, Meral Ozsoyoglu, Sunita Sarawagi, Min Wang, and Gerhard Weikum.

SIGMOD Information Director:

Curtis Dyreson, Utah State University, < curtis.dyreson AT usu.edu>

Associate Information Directors:

Manfred Jeusfeld, Georgia Koutrika, Michael Ley, Wim Martens, Mirella Moro, Rachel Pottinger, Altigran Soares da Silva, and Jun Yang.

SIGMOD Record Editor-in-Chief:

Ioana Manolescu, Inria Saclay—Île-de-France, <ioana.manolescu AT inria.fr>

SIGMOD Record Associate Editors:

Yanif Ahmad, Denilson Barbosa, Pablo Barceló, Vanessa Braganholo, Marco Brambilla, Chee Yong Chan, Anish Das Sarma, Glenn Paulley, Alkis Simitsis, Nesime Tatbul and Marianne Winslett.

SIGMOD Conference Coordinator:

K. Selçuk Candan, Arizona State University <candan AT asu.edu>

PODS Executive Committee: Rick Hull (chair), <hull AT research.ibm.com>, Michael Benedikt, Wenfei Fan, Maurizio Lenzerini, Jan Paradaens and Thomas Schwentick.

Sister Society Liaisons:

Raghu Ramakhrishnan (SIGKDD), Yannis Ioannidis (EDBT Endowment).

Awards Committee:

Rakesh Agrawal, Elisa Bertino, Umesh Dayal, Masaru Kitsuregawa (chair, University of Tokyo, <kitsure AT tk1.iis.u-tokyo.ac.jp>) and Maurizio Lenzerini.

Jim Gray Doctoral Dissertation Award Committee:

Johannes Gehrke (Co-chair), Cornell Univ.; Beng Chin Ooi (Co-chair), National Univ. of Singapore, Alfons Kemper, Hank Korth, Alberto Laender, Boon Thau Loo, Timos Sellis, and Kyu-Young Whang.

[Last updated : March 21st, 2013]

SIGMOD Edgar F. Codd Innovations Award

For innovative and highly significant contributions of enduring value to the development, understanding, or use of database systems and databases. Until 2003, this award was known as the "SIGMOD Innovations Award." In 2004, SIGMOD, with the unanimous approval of ACM Council, decided to rename the award to honor Dr. E. F. (Ted) Codd (1923 - 2003) who invented the relational data model and was responsible for the significant development of the database field as a scientific discipline. Recipients of the award are the following:

Michael Stonebraker (1992)	Jim Gray (1993)	Philip Bernstein (1994)
David DeWitt (1995)	C. Mohan (1996)	David Maier (1997)
Serge Abiteboul (1998)	Hector Garcia-Molina (1999)	Rakesh Agrawal (2000)
Rudolf Bayer (2001)	Patricia Selinger (2002)	Don Chamberlin (2003)
Ronald Fagin (2004)	Michael Carey (2005)	Jeffrey D. Ullman (2006)
Jennifer Widom (2007)	Moshe Y. Vardi (2008)	Masaru Kitsuregawa (2009)
Umeshwar Dayal (2010)	Surajit Chaudhuri (2011)	Bruce Lindsay (2012)

SIGMOD Contributions Award

For significant contributions to the field of database systems through research funding, education, and professional services. Recipients of the award are the following:

Maria Zemankova (1992)	Gio Wiederhold (1995)	Yahiko Kambayashi (1995)
Jeffrey Ullman (1996)	Avi Silberschatz (1997)	Won Kim (1998)
Raghu Ramakrishnan (1999)	Michael Carey (2000)	Laura Haas (2000)
Daniel Rosenkrantz (2001)	Richard Snodgrass (2002)	Michael Ley (2003)
Surajit Chaudhuri (2004)	Hongjun Lu (2005)	Tamer Özsu (2006)
Hans-Jörg Schek (2007)	Klaus R. Dittrich (2008)	Beng Chin Ooi (2009)
David Lomet (2010)	Gerhard Weikum (2011)	Marianne Winslett (2012)

SIGMOD Jim Gray Doctoral Dissertation Award

SIGMOD has established the annual SIGMOD Jim Gray Doctoral Dissertation Award to *recognize excellent research by doctoral candidates in the database field.* Recipients of the award are the following:

- **2006** *Winner*: Gerome Miklau, University of Washington. *Runners-up*: Marcelo Arenas, University of Toronto; Yanlei Diao, University of California at Berkeley.
- **2007** *Winner*: Boon Thau Loo, University of California at Berkeley. *Honorable Mentions*: Xifeng Yan, University of Indiana at Urbana Champaign; Martin Theobald, Saarland University
- **2008** *Winner*: Ariel Fuxman, University of Toronto. *Honorable Mentions*: Cong Yu, University of Michigan; Nilesh Dalvi, University of Washington.
- **2009** *Winner*: Daniel Abadi, MIT. *Honorable Mentions*: Bee-Chung Chen, University of Wisconsin at Madison; Ashwin Machanavajjhala, Cornell University.
- **2010** *Winner:* Christopher Ré, University of Washington. *Honorable Mentions*: Soumyadeb Mitra, University of Illinois, Urbana-Champaign; Fabian Suchanek, Max-Planck Institute for Informatics.
- **2011** *Winner*: Stratos Idreos, Centrum Wiskunde & Informatica. *Honorable Mentions*: Todd Green, University of Pennsylvania; Karl Schnaitter, University of California in Santa Cruz.
- **2012** *Winner*: Ryan Johnson, Carnegie Mellon University. *Honorable Mention*: Bogdan Alexe, University of California in Santa Cruz.

A complete listing of all SIGMOD Awards is available at: **http://www.sigmod.org/awards/**

[Last updated : December 18th, 2012]

Editor's Notes

Welcome to the June 2013 issue of the ACM SIGMOD Record!

The issue opens with a vision paper by Bartoš, Skopal and Moško on efficient indexing techniques supporting similarity search. Efficient techniques for similarity search are required in many contexts such as bioinformatics, social networks and multimedia databases. Importantly, while the most commonly known distance functions based on which similarity is assessed are related to some metric space and obey some corresponding constraints (think of the triangle inequality for distances in an Euclidian space), there are important non-metric (or unconstrained) distance functions. The authors focus on the resulting unconstrained similarity search problem, which is the target of their SIMDEX framework. SIMDEX allows a dataset-driven exploration of alternative indexing strategies in order to support efficient and scalable similarity search. The authors present experiments validating their framework, and discuss directions for future development.

The article by Montolio, Dominguez-Sal and Larriba-Pey investigates the connection between two hotly discussed metrics characterizing scientific conferences : conference quality, respectively, endogamy, defined as repeated collaborations (co-signing) of recurring sets of co-authors. The authors introduce a simple metric for endogamy and evaluate it for a set of conference and journals, including well-known database ones such as SIGMOD, VLDB, ICDE and ICDT. The finding of this study is that low endogamy (thus, time-varying co-authorship networks) correlates with conferences and journals reputed of high quality; in a time when data management research takes strong interest in social networks, this article is an interesting opposing perspective of social graph analysis applied to database publications themselves !

The survey by Guille, Hacid, Favre and Zighed keeps us in the area of social networks, more specifically focusing on information diffusion patterns. The core questions considered are: which information items are popular and diffused the most, how, why and through which paths, and which are the important influencers in the network. The authors introduce a set of basis notion related to information diffusion and then classify existing algorithms and methods for answering these questions. This clear, well-illustrated survey is very timely, given both the database community interest in social network analysis, and the spread of research in this area across several communities, including data mining, text analysis, and algorithms on graphs.

In the Systems and Prototypes column, Nakashole, Weikum and Suchanek present PATTY, a system for extracting semantic relationships out of text snippets found on the Web. The article discusses the successive extraction stages (text pattern extraction, syntactic-ontological pattern transformation, pattern generalization and subsumption and synonym mining) implemented within PATTY, describes the modules which are part of the tool, and ends by providing precision/recall results and applications.

The Distinguished Profiles column features an interview with Jeffrey Vitter, now the provost and executive vice chancellor at the University of Kansas. He talks about his PhD student days in Stanford, the lessons learned from Jeff Ullman, the importance of understanding both theory and systems in order to get good results at either of them, applying wavelets to database problems, the interest of having an MBA on top of a PhD in Computer Science, the interest of listening to problems from other disciplines, whether chemistry, physics, and music, to understand where actual open data management problems lie and investigate them.

In the Research centers column, Bressan, Chan, Hsu, Lee, Ling, Ooi, Tan and Tung give an overview of data management reseach at the National University of Singapore (NUS). The work areas surveyed in the paper include cloud-based data management, data management technologies applied to digital megacities,

for instance in the area of environment monitoring and real-time location-aware social search, data analytics, mining and visualization.

The Open forum column features a quite unique column where Graham Cormode spells out the duties, chores, and pleasures of an Associate Editor. Having served for a few years as an Associate Editor myself, and having coopted many of today's SIGMOD Record Associate Editors, I am in a position to appreciate the clear, thoughtful, and thoroughly entertaining explanations! I am sure they will clarify things for many current and future scientific journal editors and reviewers, and demystify the ways refereed journals are produced to the benefit of editors, reviewers, and authors alike.

The issue closes with two reports. First, Benedikt and Olteanu report on the first Workshop on Innovative Querying of Streams, held in Oxford in September 2012. The workshop was organized in connection to a research project on XML streams. The topics explored include social streams, semantic Web data streaming, stream uncertainty, monitoring and distribution.
Last but not least, the second report from Atzeni, Jensen, Orsi, Ram, Tanca and Torlone summarizes the discussions of a panel held in the Non-Conventional Data Access (NoCoDa) workshop 2012, on the topic of NoSQL models, querying, and overall place in the history and perspectives of data management. Read this very lively rendition of the panel's talks to form your own opinion whether conceptual database design and physical data independence really are too old for our scientific "country"?

Your contributions to the Record are welcome via the RECESS submission site (http://db.cs.pitt.edu/recess). Prior to submitting, be sure to peruse the Editorial Policy on the SIGMOD Record's Web site (http://www.sigmod.org/publications/sigmod-record/sigmod-record-editorial-policy).

Ioana Manolescu

June 2013

Past SIGMOD Record Editors:

Harrison R. Morse (1969)
Daniel O'Connell (1971 – 1973)
Randall Rustin (1974-1975)
Douglas S. Kerr (1976-1978)
Thomas J. Cook (1981 – 1983)
Jon D. Clark (1984 – 1985)
Margaret H. Dunham (1986 – 1988)
Arie Segev (1989 – 1995)
Jennifer Widom (1995 – 1996)
Michael Franklin (1996 – 2000)
Ling Liu (2000 – 2004)
Mario Nascimento (2005 – 2007)
Alexandros Labrinidis (2007 – 2009)

Towards Efficient Indexing of Arbitrary Similarity

[Vision paper]

Tomáš Bartoš Tomáš Skopal Juraj Moško

Charles University in Prague, Faculty of Mathematics and Physics, SIRET Research Group
Malostranské nám. 25, 118 00 Prague, Czech Republic
{bartos, skopal, mosko}@ksi.mff.cuni.cz

ABSTRACT

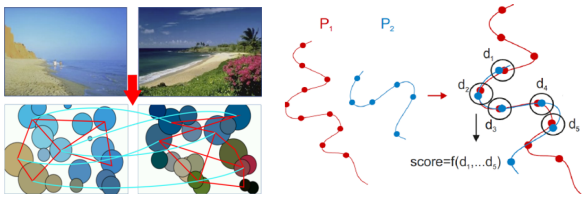
The popularity of similarity search expanded with the increased interest in multimedia databases, bioinformatics, or social networks, and with the growing number of users trying to find information in huge collections of unstructured data. During the exploration, the users handle database objects in different ways based on the utilized similarity models, ranging from simple to complex models. Efficient indexing techniques for similarity search are required especially for growing databases.

In this paper, we study implementation possibilities of the recently announced theoretical framework SIMDEX, the task of which is to algorithmically explore a given similarity space and find possibilities for efficient indexing. Instead of a fixed set of indexing properties, such as metric space axioms, SIMDEX aims to seek for alternative properties that are valid in a particular similarity model (database) and, at the same time, provide efficient indexing. In particular, we propose to implement the fundamental parts of SIMDEX by means of the genetic programming (GP) which we expect will provide high-quality resulting set of expressions (axioms) useful for indexing.

1. INTRODUCTION

The content-based retrieval is widely used in various areas of computer science including multimedia databases, data mining, time series, genomic data, social networks, medical or scientific databases, biometric systems, etc. In fact, searching collections of a priori unstructured data entities requires a kind of aggregation that ranks the data as more or less relevant to a query. A popular type of such a mechanism is the *similarity search* where, given a sample query object (e.g., an image), the database searches for the most similar objects (images). Two unstructured objects represented by their descriptors are compared by a similarity function, which produces a single numerical score interpreted as the degree of similarity between the two original objects.

For a long time, the database-oriented research



(a) Image similarity (b) Protein similarity

Figure 1: Sample similarity models

of similarity search employed the definition of similarity restricted to the *metric space* model with fixed properties of *identity*, *positivity*, *symmetry*, and especially *triangle inequality*, using *metric access methods* for indexing [2, 20, 14].

Together with the increasing complexity of data types across various domains, recently there appeared many similarities that were not metric – we call them *nonmetric* or *unconstrained* similarity functions [17]. As the nonmetric similarity functions are not constrained by any properties that need to be satisfied (unlike the metric ones), they allow to better model the desired concept of similarity and therefore lead to more precise retrieval (see Fig. 1a for a robust matching using local image features).

Also nonmetric similarities allow to design models that cannot be formalized into a closed-form equation. They could be defined as heuristic algorithms such as an alignment or a transformational procedure, while the enforcement of metric axioms could be very difficult or even impossible. As an example (see Fig. 1b), consider alignment algorithms for measuring functional similarity of protein sequences [18] or structures [8].

However, usually just the *database experts* are concerned with the existence of specific properties in a similarity function, as the properties enable the ways how to index the database for efficient similarity search. But database experts usually do not investigate the applicability of their techniques to specific domains. On the other hand, there are much

larger *domain expert* communities of different kinds – people who use specialized similarity search applications and are ready to apply any method in order to get expected results. These experts typically do not care about the indexing techniques or performance issues to a certain extent, so enforcement of any indexing-specific properties in their similarity functions is out of their expertise. For them, the best approach is to use the simplest (possibly inefficient) database methods as they are easy to implement. However, in long term and with large-scale databases, the efficiency will become a critical factor for choosing suitable similarity search methods.

Based on the different interests of database and domain research communities, the main goal of our research is to find a complex solution that provides the various domain experts with a database technique that allows effective similarity search yet that does not require any database-specific intervention to the generally unconstrained similarity models. In the following text, we shortly summarize previous attempts to unconstrained (nonmetric) similarity search before we sketch the idea of how to apply genetic programming for this purpose.

2. MOTIVATION

It is not always easy for domain experts to invent a perfect similarity measure, mostly represented as a distance (dissimilarity) function δ, and use it efficiently for large-scale databases with no compromise. The general way how to efficiently search is to use the *lowerbounding* principle – instead of computing expensive distances between a query object and all database objects a cheaper lowerbounding function LB is applied to filter the irrelevant ones.

The first lowerbounding approach might be to meet requirements of the *metric space model* by modifying the similarity model. Then a lowerbound function $LB_\triangle$ utilizing the *triangle inequality* is used

$$\delta(q,o) \geq LB_\triangle(\delta(q,o)) = |\delta(q,p) - \delta(p,o)| \quad (1)$$

for query q, pivot (reference) object p, and database object o. However, such a transformation might spoil the benefits of the original model.

So, the next option is to use an indirect variation of the model leveraging the known mapping approaches such as TriGen [15] which "converts" the nonmetric similarities into metric ones and, again, the metric model might be used. However, this is not always the best-case scenario as it might lead to either large retrieval error or low indexability [17].

Hence, there appeared some alternative methods of database indexing for unstructured data, such as the *Ptolemaic Indexing* [9, 11]. Here, the *Ptolemy's inequality* is used to construct lowerbounds. It states that for any quadrilateral, the pairwise products of opposing sides sum to more than the product of the diagonals. So, for any four database objects x, y, u, $v \in \mathcal{D}$, we have:

$$\delta(x,v)\cdot\delta(y,u) \leq \delta(x,y)\cdot\delta(u,v)+\delta(x,u)\cdot\delta(y,v) \quad (2)$$

For Ptolemaic lowerbounding LB_{ptol} with a given set of pivots $\mathbb{P}$, the bound δ_C derived from (2) is maximized over all pairs of distinct pivots [9, 11]:

$$\delta(q,o) \geq LB_{\text{ptol}}(\delta(q,o)) = \max_{p,s\,\in\,\mathbb{P}} \delta_{\text{C}}(q,o,p,s) \quad (3)$$

The ptolemaic indexing was successfully used with the *signature quadratic form distance* [11] that is suitable for effective matching of image signatures [1]. The idea of ptolemaic indexing shows that finding new indexing axioms could be a solution to speed-up similarity search in other way than mapping the problem to the metric space model.

3. RELATED WORK

We acknowledge that "lowerbounding problem" has been studied widely from various perspectives but as we found out this is true mostly for specific domains such as text or information retrieval (IR). For example, the recent paper [4] discusses axioms or constraints useful for term-weighting functions but it is limited to IR, while in [12] authors try to overcome improper lowerbounds with a new sufficiently large lowerbound for term frequency normalization (hardly applicable outside IR area).

Another work [13] reveals dynamic pruning strategies based on upper bounds to quickly determine the dissimilarity between an object and a query and thus quickly filter out objects; again designed for IR domain only.

Next, the definitions of axioms and constraints for similarity functions used in text retrieval systems are studied in [7], but the author provides only the theoretical background.

Interestingly, there exists a framework that provides an axiomatic approach for developing retrieval models [6]. It searches the spaces of candidate retrieval functions with the aim of finding the one that satisfies specific constraints. Although our approach might look the same, there are significant differences from our work. Particularly because authors are strongly connected to IR as they assume "bag-of-terms" representation of objects and they create retrieval functions inductively with respect to specific retrieval criteria. Most importantly, they focus on modeling the relevance rather than developing efficient database indexing techniques.

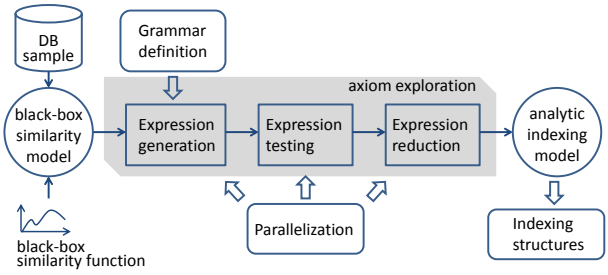


Figure 2: SIMDEX Framework high-level overview

So, a general method that provides a correct lowerbound for any domain has not been identified yet. And here we see the great potential for our research – to create and deliver a dataset-driven framework that is able to find lowerbounds for any given similarity space. This will then result in the efficient indexing method applicable to any domain.

4. SIMDEX FRAMEWORK

Our work outlines an alternative approach to similarity indexing motivated by the Ptolemaic indexing. Instead of "forcing" the distance and/or data to comply with the metric space model, for some datasets it could be more advantageous to employ completely different indexing model that provides cheap construction of lowerbounds. We intend to replace expensive distance computations between all pairs of objects by a cheaper lowerbounding function that filters out the non-interesting objects.

Therefore our major research goal is to develop a robust algorithmic framework for dataset-driven automatic exploration of axiom spaces for efficient and effective similarity search at large scale. We already described the SIMDEX framework and sketched a high-level overview (see Fig. 2) of the framework's stages (the inner components) in [16]. In that preliminary study, we designed only the theoretical concept while in this work, we verified our thoughts and clarify our vision with future steps.

4.1 Concept of SIMDEX Framework

As the input we consider a distance matrix for a *database sample* (S) computed with a *black-box distance function* (δ). This matrix consists of a set of values obtained by computing pair-wise distances between objects in the sample – it is our "mining field". The resulting output is a set of expressions (so called *axioms*) valid in the given similarity space that might be used for effective similarity search.

Using the basic idea of iteratively constructing and testing the expressions against the distance matrix, we are able to algorithmically explore axiom spaces specified in a syntactic way. This approach does not use a single canonized form and a tuning parameter, as other mapping approaches or the algorithm *TriGen* do. As the result, we will be able to discover the existing lowerbounding forms such as triangle inequality (Eq. 1) or Ptolemy's inequality (Eq. 3) as two instances in the axiom universe.

Moreover, since the resulting set of axioms (analytical properties) will be obtained in their lowerbounding forms, they can be immediately used for filtering purposes in the same way as ptolemaic indexing was implemented [11].

4.2 Framework Overview

In this section, we briefly introduce and describe the framework stages but for more details about particular components, we refer readers to our initial study in which the architecture and the methodology are described properly [16].

As the initial step, we use the grammar theory to create a *grammar definition* G based on which the expressions are subsequently generated. The generated expressions are in the standardized form of $\delta(q, o) \geq LB$, where LB will be expanded to various forms. Expressions cannot be computationally too expensive to evaluate and always include $\delta(\cdot, p)$, where pivot p is a fixed reference point.

Because the grammar-based generating of expression leads to an infinite universe, we limit the set of tested inequalities by (a) using the signatures of expressions that exclude various forms of the same expression (i.e., *fingerprints*), and (b) discarding meaningless expressions such as $\frac{x}{x}$, $-x$, ...

After we generate candidate expressions, they are tested against the precomputed distance matrix. As we require 100% precision, only such expressions are valid for which all tests are evaluated as `TRUE`.

To further condense the number of expressions we could refine the result by discarding weaker expressions or combining expressions into a compound expression, so only the best expressions will remain.

The last (indexing) step directly verifies the feasibility of the resulting set of expressions/axioms in practice within sample indexing tasks and validates the filtering power of each expression. We focus on the pivot table [2, 20] as it could be immediately used as an indexing structure for any kind of lowerbound expressions that involve pivots.

Although we optimize all stages, the exhaustive computation is still in place. Therefore, we assume massive parallelization of the exploration process leveraging classic multi-core CPU systems with multi-threading. For the future, we consider MapReduce technique [5] applied to a CPU farm or to a supercomputer architecture with lots of cores.

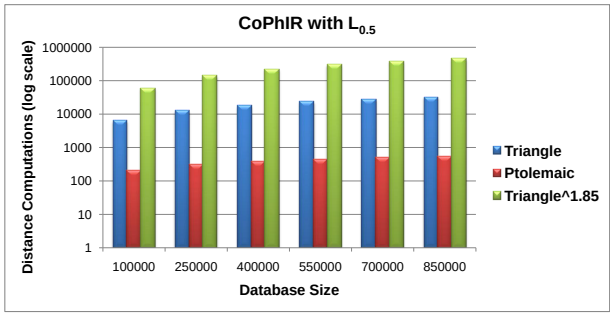


Figure 3: CoPhIR - Distance computations (log scale)

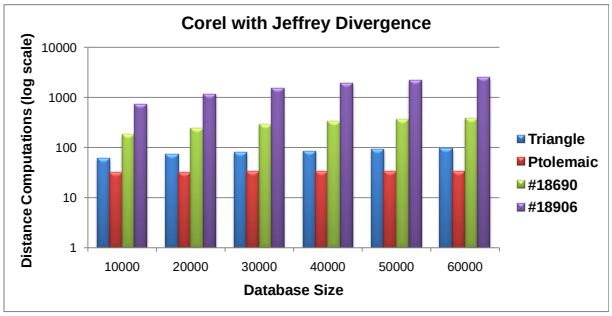


Figure 5: Corel - Distance computations (log scale)

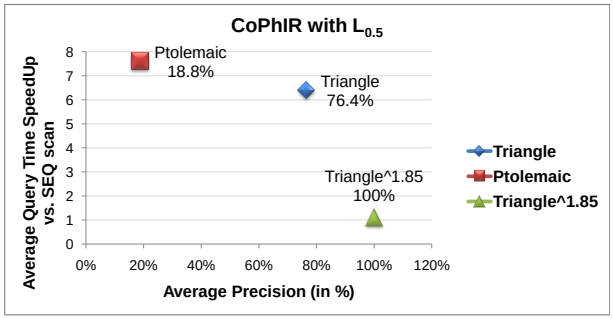


Figure 4: CoPhIR - Avg speedup vs. avg precision

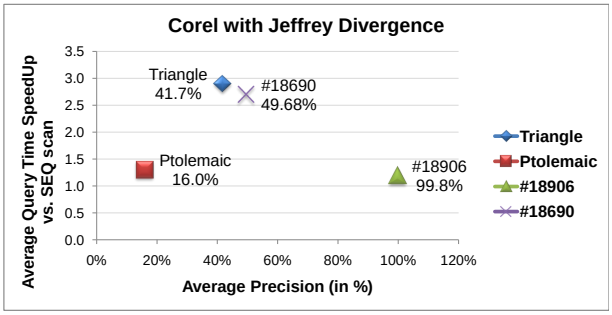


Figure 6: Corel - Avg speedup vs. avg precision

4.3 Preliminary results

After the naive implementation of all individual framework stages, we applied the prototype to the real-world datasets focusing on nonmetric similarity models in which metric postulates used for indexing and querying produced notable errors. This step validates our theoretical concept and as a proof we present convincing preliminary results.

Using a sample database (consisting of 25 objects), we tested CoPhIR[1] dataset with nonmetric $L_{0.5}$ distance and color histograms from *Corel Image Features*[2] dataset using nonmetric Jeffrey Divergence distance measure [17]. We verified the outcomes (resulting axioms) on indexing processes with Pivot Table [20] while studying the precision compared to results of sequential scan (SEQ), number of distance $\delta(\cdot,\cdot)$ computations (DCs) as the basic efficiency measure, and average speedup.

The best result for CoPhIR was the expression

$$\delta(q,o) \geq Triangle^{1.85}(\delta,q,p,o) = |\delta(q,p)-\delta(p,o)|^{1.85}$$

which does not dominate in number of DCs (Fig. 3) but it clearly produces no errors (Fig. 4) together with 1.1× speedup vs. SEQ scan.

[1]http://cophir.isti.cnr.it/
[2]http://goo.gl/SaOms

For Corel, we found the following expressions

$$\#18690 \quad \delta(q,o) \geq Triangle^{2}(\delta,q,p,o) = |\delta(q,p) - \delta(o,p)|^2$$
$$\#18906 \quad \delta(q,o) \geq (\delta(q,p_1) - \delta(o,p_1)) \cdot (\delta(q,p_2) - \delta(o,p_2))$$

While the squared triangle inequality (#18690) is only slightly more precise than triangle $LB_{\triangle}$ (Fig. 6), we achieved an enormous success with the next expression (#18906) – 99.8% precision together with 1.2× speedup compared to sequential scan. Although $LB_{\triangle}$ still dominates in the number of DCs (Fig. 5), it produces notable error rates (up to 59%).

4.4 Challenges

With the implemented prototype, we verified the feasibility of our concept; however, there appeared few issues that we need to overcome in order to provide a real and viable end-to-end solution. Namely, we need to address following challenges:

- **Expression Generation** – The basic concept of generating expressions iteratively covers all expressions (which is the advantage), however, a complex axiom valid in the given space could take enormous time to be revealed.
- **Expression Similarity** – Despite using the fingerprinting, we still struggle with testing only unique expressions and skipping the various forms of the similar ones, as there are infinite forms of how to express a single math expression.

- **Expression Testing** – We have to compromise between a large number of expressions to be tested and a bigger sample size. Testing the whole sample does not have to be always appropriate and we might take only some interesting objects from the sample.
- **Verifying indexing model** – To validate that resulting axioms could be used for indexing purposes, we run a separate indexing process on the data outside the sample which is correct but time-consuming.

5. GENETIC PROGRAMMING VISION

In order to improve and extend the framework capabilities and to overcome mentioned challenges (see Section 4.4), we propose using genetic programming (GP) as the main driver of generating and testing expressions. The concept of GP is not new and has been studied for several years since one of the first inspiring books was published [10]. In general, GP applies evolutionary patterns to a particular problem to achieve a specific goal using operations such as selection, crossover, or mutation [3].

We expect that GP-based approach will give the real power to the purely theoretical SIMDEX Framework (i.e., it will "materialize the theory"), will boost the efficiency of axiom discovery and speedup the axiom exploration process. Applying the principles of natural expression evolution will then lead to faster axiom resolution. Maybe we will not find all axioms valid in the given space but this is not our primary goal. In the first phase, we concentrate on detecting at least some axioms that will increase the efficiency of the indexing/filtering process.

5.1 GP-based SIMDEX Framework

Using GP-based method within the axiom exploration requires several customizations of individual framework stages. For this purpose, we propose and design the next generation of SIMDEX Framework (Fig. 7) which is how we perceive our future research. Connecting the existing theoretical concept together with GP-based algorithms (which will enrich it with the real and applicable context) we will gain a powerful tool for axiom exploration.

Our vision and the real motivator is, that given arbitrary user-defined similarity space, we will be able to find valid axioms within a *reasonable* and *acceptable* time frame. And we strongly believe GP-based components will help us to achieve this. Essentially, the novel GP-based axiom exploration process will address highlighted challenges with

- **Initial Population** - After we create the initial population with the existing expression

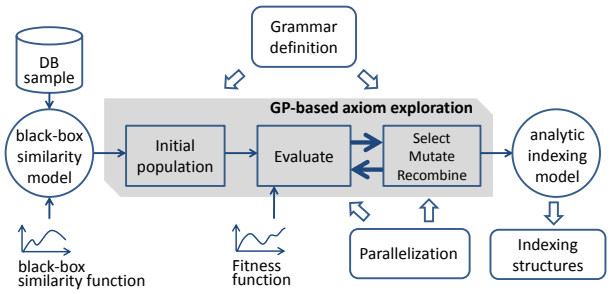


Figure 7: GP-based SIMDEX Framework

generator, additional expressions will be generated by the evolution algorithms which we expect will lead to "good" axioms early enough. We will consider two variants: *iteratively* and *randomly* built sets.

- **Evaluate** - This stage partially corresponds to Expression Testing, however we need to take into account several *fitness functions* to choose from such as (a) complete testing of a smaller distance matrix, (b) sampling n-tuples from a medium distance matrix, or (c) imitating a pivot-based search on a large distance matrix, which will give us better scalability of results.
- **GP-based operations** (Select, Mutate, Recombine) - Based on the evaluation results, we will *select* the most promising expressions and add them to the next generation. Some of them will be modified (*mutated*) or *recombined* with others (i.e., the crossover of expression trees) in order to boost their efficiency and find better expressions. During this stage, we need to test expression similarities and for this purpose, we consider applying a similarity measure to find similarities in expression trees (e.g., tree edit distance [15]) together with our previously proposed fingerprinting method.

We see the great potential in creating multiple generations of expressions based on the feedback from the evaluation, so we can try to modify the expressions to improve their efficiency accordingly. Depending on results, we will handle the mutation and recombination processes either in a completely random way, or there will be some logic behind to improve specific parts of an expression (modifying specific nodes in the expression tree).

The availability of multiple fitness functions gives us the opportunity to study expressions' behavior in different testing environments and potentially to come up with special characteristics of expressions and their suitability for specific datasets.

Another advantage is that GP has been studied and applied widely to lots of different areas and

there exists multiple options of how to perform each operation – sampling, recombination, or mutation, in order to obtain the next generation [19]. Therefore we can pick the method that will be mostly related and suitable to mathematical expressions.

6. CONCLUSION AND FUTURE WORK

With the preliminary implementation of purely theoretical **SIMDEX** Framework, we are able to demonstrate how to deal with the efficiency of similarity search in nonmetric spaces in other way than forcing the domain experts to implant and use metric postulates in their similarity models. Based on the results, we conclude that our framework is capable of finding alternative ways of indexing that speed up high-precision similarity queries.

However, to achieve this within an acceptable time frame and to find interesting axioms, we need to optimize it dramatically. For this purpose, we push our framework towards evolutionary algorithms (e.g., genetic programming). Doing so, we expect to explore the search space of all possible expressions more effectively and to have good results quickly. This method could provide better outcomes in terms of query efficiency/effectiveness for complex nonmetric similarity models. In the metric spaces, our solution will just provide a solid alternative to qualitatively dominating state-of-the-art techniques.

7. ACKNOWLEDGMENTS

This research has been supported by Grant Agency of Charles University (GAUK) projects 567312 and 910913 and by Czech Science Foundation (GAČR) project 202/11/0968.

8. REFERENCES

[1] C. Beecks, M. S. Uysal, and T. Seidl. Signature quadratic form distance. In *Proc. ACM International Conference on Image and Video Retrieval*, pages 438–445, 2010.

[2] E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín. Searching in metric spaces. *ACM Comp. Surveys*, 33(3):273–321, 2001.

[3] N. L. Cramer. A representation for the adaptive generation of simple sequential programs. In *Proc. of the 1st Int. Conf. on Genetic Algorithms*, pages 183–187. L. Erlbaum Associates Inc., USA, 1985.

[4] R. Cummins and C. O'Riordan. An axiomatic comparison of learned term-weighting schemes in information retrieval: clarifications and extensions. *Artif. Intell. Rev.*, 28:51–68, 2007.

[5] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. In *Proc. of the 6th conf. on Symp. on Oper. Systems Design & Impl.*, USA, 2004.

[6] H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. In *SIGIR*, pages 480–487. ACM, 2005.

[7] R. K. France. Weights and Measures: an Axiomatic Approach to Similarity Computations. Technical report, 1995.

[8] J. Galgonek, D. Hoksza, and T. Skopal. SProt: sphere-based protein structure similarity algorithm. *Proteome Science*, 9:1–12, 2011.

[9] M. L. Hetland. Ptolemaic indexing. `arXiv:0911.4384 [cs.DS]`, 2009.

[10] J. R. Koza. *Genetic programming*. MIT Press, Cambridge, MA, USA, 1992.

[11] J. Lokoč, M. Hetland, T. Skopal, and C. Beecks. Ptolemaic indexing of the signature quadratic form distance. In *Similarity Search and Applications*, pages 9–16. ACM, 2011.

[12] Y. Lv and C. Zhai. Lower-bounding term frequency normalization. In *Proc. of the 20th ACM Int. Conf. on Information and knowledge management*, CIKM '11, pages 7–16, New York, NY, USA, 2011. ACM.

[13] C. Macdonald, N. Tonellotto, and I. Ounis. On upper bounds for dynamic pruning. In *Proc. of the 3rd Int. Conf. on Advances in information retrieval theory*, ICTIR'11, pages 313–317. Springer-Verlag, 2011.

[14] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann Publishers Inc., USA, 2005.

[15] T. Skopal. Unified framework for fast exact and approximate search in dissimilarity spaces. *ACM Transactions on Database Systems*, 32(4):1–46, 2007.

[16] T. Skopal and T. Bartoš. Algorithmic Exploration of Axiom Spaces for Efficient Similarity Search at Large Scale. In *Similarity Search and Applications*, LNCS, 7404, pages 40–53. Springer, 2012.

[17] T. Skopal and B. Bustos. On nonmetric similarity search problems in complex domains. *ACM Comp. Surv.*, 43:1–50, 2011.

[18] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147:195–197, 1981.

[19] D. Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.

[20] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search: The Metric Space Approach*. Advances in Database Systems. Springer-Verlag, USA, 2005.

Research Endogamy as an Indicator of Conference Quality

Sergio Lopez Montolio, David Dominguez-Sal, Josep Lluis Larriba-Pey
DAMA-UPC
Universitat Politecnica de Catalunya, Barcelona Tech
Barcelona
{slopez,ddomings,larri}@ac.upc.edu

ABSTRACT

Endogamy in scientific publications is a measure of the degree of collaboration between researchers. In this paper, we analyze the endogamy of a large set of computer science conferences and journals. We observe a strong correlation between the quality of those conferences and the endogamy of their authors: conferences where researchers collaborate with new peers have significantly more quality than conferences where researchers work in groups that are stable along time.

1. INTRODUCTION

Social sciences define endogamy as "the custom of marrying only within the limits of a local community, clan, or tribe"[1]. We can extend this concept to measure the degree of collaboration between persons. In the context of scientific publications, we consider endogamy as the inclination of a person or a group to usually collaborate (i.e., publish papers) within a small group of selected people.

Coauthorship networks represent authors as nodes in a graph and edges linking people who coauthor a paper. They provide information about how the researchers cooperate to produce new ideas [11]. It is known that not all collaborations have an equal impact, and some of them produce higher research impact [2]. Furthermore, Guimerà et al. studied a small set of journals and found that endogamy is a significant factor in the performance of research teams in some research fields such as social psychology or ecology [6]. The collaborations with new researchers open new streams of ideas, and hence are a positive indicator of good research.

In this paper, we go further in the study of the endogamy in computer science collaborations. We apply this endogamy to calculate the endogamy of a broad spectrum of computer science conferences (926) and journals (317). We observe that there is a strong influence of the endogamy of the research teams publishing in a conference on the quality of such conference (up to 80% agreement with the ERA conference ranking[2]). This shows the social importance of conferences for computer scientists, where they are able to meet new peers that in turn lead to better publications. In particular, reputed conferences such as PODS, ICDT, SIGMOD, VLDB or ICDE stand out among database conferences as having particularly low endogamy. Although this collaborative strategy works well for conferences, it is not universal, because we found that computer science journals are not affected by endogamy alike.

The correlation found between the endogamy and the quality of conferences opens the possibility to consider having metrics to evaluate the quality of a conference that are based on the social aspects of research. Currently, the evaluation of conferences relies mostly on measures based on the citations: h-index, cites per paper, pagerank, etc. [1, 5] and in few occasions (e.g. program committee relations [14]) personal relations are analyzed. But, the extraction of cites is not an easy task [3] and error free citation collection requires a large manual effort. Furthermore, the median age of citation is several years (e.g. the median age for TODS is over 10 years [13]), which delays the release of reliable qualifications for conferences and journals. In contrast, coauthor networks are easy to obtain and they describe the current information without delay. Although social metrics cannot be used to evaluate the content of an article because scientific excellence is determined by article's content and not by authors' profiles, social metrics can be computed to obtain early estimates of the quality of recent conferences.

We define the endogamy in Section 2. Then, we describe the experimental environment in Section 3. After computing the endogamy for all the available

[1] http://oxforddictionaries.com/definition/endogamy

[2] Previously known as CORE. Available at http://www.arc.gov.au/era/era_2010/archive/default.htm

journals and conferences in our dataset, we evaluate the results for conferences in general in Section 4, and for database conferences in Section 5. Finally, we evaluate analyze the endogamy of journals in Section 6.

2. ENDOGAMY COMPUTATION

Research is based on the proposal and study of new ideas. The collaboration with researchers external to the usual research team is a very good means to introduce such new ideas and allow merging the expertise from multiple fields. In this paper, we quantify this degree of new collaborations by means of a new indicator called endogamy.

We compute the endogamy of a set of authors as the inclination of a person or a group to usually collaborate (i.e., publish papers) within a small group of selected people as:

$$Endo(A) = \frac{|d(A)|}{|\bigcup_{a \in A} d(\{a\})|}, \quad (1)$$

where A is a set of authors, and $d(A)$ is the set of papers that were published by the *full* set of authors, in other words, papers coauthored by all the members of A. For example, consider the endogamy of a group formed by authors x and y, who have individually published three papers ($d(\{x\}) = \{a, b, c\}$ and $d(\{y\}) = \{b, c, d\}$). Since they have collaborated in half of their publications their endogamy, $Endo(\{x, y\})$, is: $^2/_4 = 0.5$

Endogamy of a paper: Let $A(p)$ be the set of authors of a paper p and $L_i(p) = \mathcal{P}_i(A(p))$ be the power set of authors of size i (the set of all subsets with size i within $A(p)$). Then, $L(p) = \bigcup_{i=2}^{i=|A|} L_i$ is the set of all the subsets with more than one author. We compute the endogamy of a paper p, as the aggregation of the endogamies of $L(p)$. We test several endogamy aggregations:

- `Max`: Maximum of the endogamies of all groups:

$$Endo(p) = max_{x \in L(p)}(Endo(x))$$

- `Min`: Minimum of the endogamies of all groups:

$$Endo(p) = min_{x \in L(p)}(Endo(x))$$

- `Med`: Median of the endogamies:

$$Endo(p) = med_{x \in L(p)}(Endo(L_i))$$

- `Avg`: Arithmetic mean of the endogamies:

$$Endo(p) = \frac{\sum_{x \in L} Endo(x)}{|L|}$$

	Conferences	Journals
A/A*	223	122
B	308	87
C	395	108
Total	926	317

Table 1: Conferences and journals by tier.

- `Harm`: Harmonic mean of the endogamies within $L(p)$:

$$Endo(p) = harm(\{Endo(x) | x \in L(p)\}),$$
$$where\ harm(X) = \frac{|X|}{\sum_{x \in X} \frac{1}{x}}$$

- `Avg size`: Arithmetic mean of the endogamies of the subsets of authors grouped by size:

$$Endo(p) = \frac{1}{|A| - 1} \cdot \sum_{i=2}^{i=|A|} \frac{\sum_{x \in L_i(p)} Endo(x)}{|L_i(p)|}$$

- `Harm size`: Harmonic mean of the endogamies of the subsets of authors grouped by size:

$$Endo(p) = harm(\{harm(L_i(p)) | 2 \leq i \leq |A|\})$$

Endogamy of a conference/journal: Let C be the set of articles published in a conference or a journal. We compute the endogamy as the average endogamy of its papers:

$$Endo(C) = \frac{1}{|C|} \sum_{p \in C} Endo(p) \quad (2)$$

Endo must not be seen as an absolute value of the research quality of a group of people. Indeed, the quality of an individual paper cannot be computed by simply stating the persons who wrote it. High quality research relies on good scientific content, which can be potentially written by any person. *Endo* should be seen as a probability distribution of the quality of a paper. The *Endo* value associated to a group is a number between 0 and 1. An *Endo* value close to 1 indicates that the paper is not likely to bring new ideas because the authors are not working with other members of the community. Values close to 0 show that the researchers constantly collaborate with new researchers, and thus they are more likely to introduce new ideas.

3. EXPERIMENTAL ENVIRONMENT

In order to study the influence of the endogamy of authors on the quality of conferences and journals, we rank the computer science conferences and

journals available in the DBLP database[3] by their *Endo* value[4]. In order to verify the quality of the ranking, we take the quality indicators published by the project Excellence in Research for Australia (ERA) as reference. We take the ERA evaluation performed in 2010, which ranks conferences and journals in three categories: A, B and C. In this classification, publications in category A are better than publications in category B, and publications in category B are better than publications in category C. Since the titles in DBLP and ERA are not normalized, we only select those conferences and journals that appear in both datasets with exactly the same title or acronym. After this process, we retrieve 926 conferences and 317 journals that belong to all the three ranks of ERA as shown in Table 1.

We report the degree of similarity between the ERA and *Endo* rankings by means of the agreement between both series. Given the two rankings, a pair of conferences c_1 and c_2 is concordant if $c_1 > c_2$ for both rankings (and by symmetry $c_1 < c_2$ for both rankings). Otherwise, the pair is discordant. We compute for all pairs of conferences (or journals) in the dataset, the number of concordant pairs p, and the number of discordant ones f (ties are not considered). The following percentage ratio computes the *agreement* between both rankings:

$$\rho = 100 \cdot \frac{p}{p+f} \tag{3}$$

We verify the statistical significance of our results by means of the Kendall tau [12], which is a non parametric test that measures the rank correlation between two lists without making assumptions of the sorting method, and ANOVA, which is suited for comparing different configurations of our metric using the R statistical package[5].

4. CONFERENCE ANALYSIS

We ranked the conferences using the six described variants of *Endo*. In this first experiment, we removed entities with low activity: those conferences with less than 500 papers in all their history. With this, we ended up with a total of 241 conferences to be used for the first experiment. We show later that the conclusions are the same if no cleanup is performed. The dark series of Figure 1 shows the agreement for each aggregation technique. We observe that the ranking of conferences performed by *Endo* has a very strong agreement with those of ERA independently of the aggregation performed. By means of the Kendall Tau coefficient test, we found that such correlations are statistically significant for all the aggregation techniques. Among them, `Max` and `Avg` are the best aggregation techniques. This corresponds to selecting the most endogamous group of authors, or average endogamy of all subsets of authors, respectively.

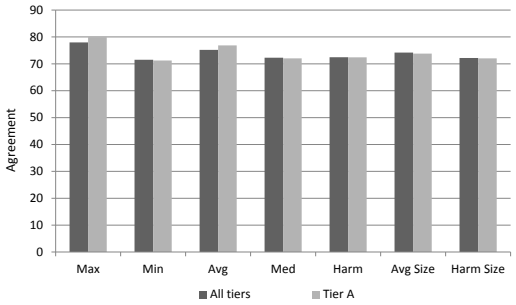


Figure 1: Agreement ρ for conferences with more than 500 papers.

We also consider the case of deciding whether a conference is a top tier (A) or a non top tier conference (B and C) according to ERA. We depict the agreement with this binary decision in the light series of Figure 1 showing that it also correlates well, being the influence statistically significant considering the Kendall coefficient.

We observed that depending on the conference tier, the distribution of *Endo* changes. We illustrate this change as a boxplot in Figure 2 for the conferences in the previous experiment, where we depict *Endo* using `Avg` with respect to the ERA tier. Note that the median *Endo* increases as we lower the conference quality, and the median *Endo* of a tier is lower than the first quartile of the next ranked tier consistently.

We verify the significance of the differences by means of an ANOVA test. We first performed a random sample of 50 conferences of each tier, adding up 150 conferences in total and compared their *Endo* in logarithmic scale. The ANOVA allows us to conclude that there exists statistically significant differences between the three tiers considered with respect to *Endo*. In order to improve the confidence of our statistical analysis, we applied resampling. We selected ten new samples, where each sample contains 50 conferences in each tier, and recomputed the ANOVA procedure. In all the cases, the results showed significant differences between tiers,

[3] `http://www.informatik.uni-trier.de/~ley/db`

[4] When we compute of a paper p using Equation 1, we consider only collaborations performed before the publication date of p. So, we do not introduce unavailable information about subsequent collaborations after p was published.

[5] All statistical test in the paper are performed with confidence level $\alpha = 0.05$

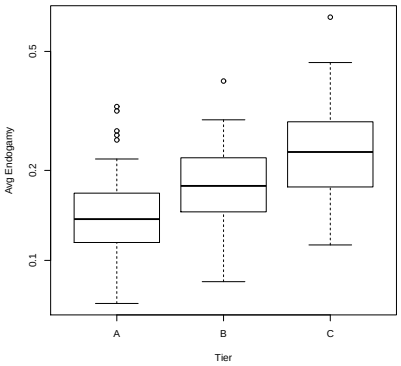


Figure 2: *Endo* **per conference tier using** `Avg`**.**

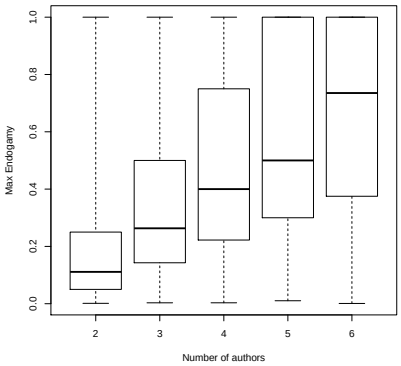


Figure 3: *Endo* **using** `Max` **vs. authors of a paper.**

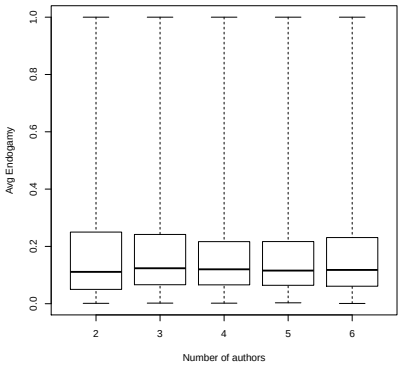


Figure 4: *Endo* **using** `Avg` **vs. authors of a paper.**

and thus, we conclude that each conference tier has a characteristic *Endo*. The different pairs of tiers have been compared using a Tukey's test, concluding that for any pair of tiers their *Endo* is statistically different.

Impact of parameters in *Endo***:** We observed that `Max` and `Avg` are the best candidates to be considered as quality indicators of conferences. After verifying the significance of their predictions (we showed in the previous section the results for `Avg` and for space reasons we do not report those for `Max`), we proceed to analyze with more detail the impact of the variables involved in the computation of *Endo*.

First, we analyze the impact of the number of authors in the computation of the endogamy of a paper. We separate the papers in groups by the number of authors and plot *Endo* for each paper in the group as a boxplot in Figures 3 and 4. We expected that the number of authors would not be relevant for the quality of the paper. We found that despite the higher precision of `Max`, the value of *Endo* obtained with it depends on the number of authors of a paper: more authors imply larger *Endo*. `Max` takes into account only the most endogamic group, and with more authors there are more subgroups that may have large endogamy. On the other hand, Figure 4 shows an homogeneous distribution of endogamies for `Avg` no matter the number of authors. We conclude that `Max` gives biased results between conferences with different distributions of authors but this is not the case for `Avg`. Therefore, in the following experiments we focus on `Avg`.

In our next experiment, we study if the number of papers of a conference and the number of papers per author have an impact in the accuracy of *Endo* as a predictor. We set five levels for each variable:

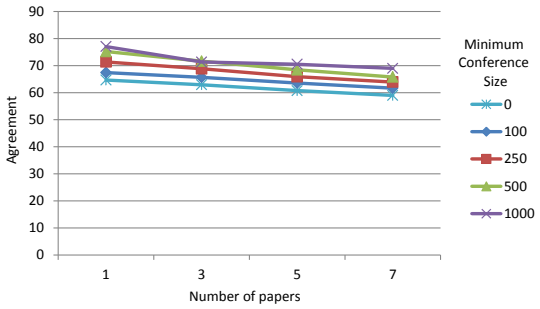


Figure 5: Agreement ρ **for** *Endo* **using** `Avg`**. Series are the minimum count of papers of a conference. The X-axis is the minimum number of papers for a group of authors.**

we study conferences with any number of papers and a minimum of 100, 250, 500 and 1.000 papers; and filter groups of authors with at least 1, 3, 5 and 7 papers. This produces twenty configurations in a full factorial design, which are plotted in Figure 5. We observe a defined trend for each variable. First, we observe that considering authors with few papers (novel authors) improves the accuracy of *Endo*. This result suggests that the impact of non experienced researchers in research teams is not negligible. Since people who publish for the first time reduce the endogamy of the research team, these results suggest that the inexperience of new researchers is overcome by the novelty of ideas that they can provide. With respect to conference size, we see that for conferences with a large number of papers the agreement is larger.

Both trends indicate that the more observations are taken into account (and thus the endogamy of more papers and more authors), the better the pre-

Conference	Tier	Avg. Endo
PODS	A	0.083
ISIT	B	0.085
EDBT	A	0.095
ICDE	A	0.108
PKDD	A	0.120
SIGMOD	A	0.122
MDM	C	0.126
ASIACRYPT	A	0.132
DASFAA	A	0.132
PAKDD	A	0.133

All history

Conference	Tier	Avg. Endo
PODS	A	0.058
CRYPTO	A	0.062
ICDT	A	0.065
DBLP	B	0.065
SIGMOD	A	0.073
EUROCRYPT	A	0.073
VLDB	A	0.077
EDBT	A	0.079
ASIACRYPT	A	0.080
ICDE	A	0.081

Years 2003-2012

Figure 6: "Data Format" conferences in ERA with the lowest *Endo*.

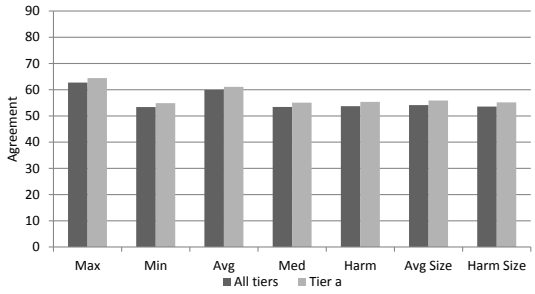


Figure 7: Agreement ρ for journals with more than 100 papers

diction power of *Endo*. As more papers are aggregated, the trends for *Endo* are stronger as a consequence of the law of large numbers.

5. DATABASE CONFERENCES

For this section, we focus on the set of conferences marked as "Data Format" in the ERA list. We computed the *Endo* value of all these conferences and ranked them. In Figure 6, we report the top 10 conferences in terms of *Endo*. We computed two result sets: the one on the left considers all the editions performed by the conferences, and the one on the right only accounts for the last ten years. We find that both lists contain a majority of conferences of excellence: on the left and on the right, 8 and 9 out of the 10 conferences classified belong to tier A, respectively. *Endo* is able to distinguish the most relevant conferences in the area: PODS, ICDE, SIGMOD, EDBT, VLDB, ICDT... Most of them appear in both lists showing the correlation of *Endo* and the quality of database conferences.

We found that time is a relevant factor in computing the endogamy of database conferences, as can be seen comparing both lists. In absolute terms, the endogamy of the latest years is considerably smaller than twenty or thirty years ago. The reason is that the database field has been a popular one and the number of authors has grown in the latest years, which provides a potentially larger number of collaborations. For example, the number of different authors that have published in SIGMOD in the last decade (2003-2012) is 2,349 compared to 1,465 in the previous decade (1993-2002).

On the other hand, conferences (and in particular those considered as the best) tend to have a worse *Endo* in the first editions and reduce their *Endo* along time. One example is VLDB that in the first five editions had endogamies above 0.4, which is significantly larger to the average of the latest ten years, 0.077. For this reason, VLDB is classified in the 11th position in the left list and does not appear in the list. We detected similar patterns for SIGMOD, ICDE, EDBT or DASFAA, just to mention a few. This pattern seems more correlated to the longevity of the conference rather than the exact year because conferences starting in 70's, 80's and 90's show such a lowering trend, as discussed with more detail in [8]. According to these results, the evaluation of a recent window of years provides more accurate tier predictions by *Endo*.

6. JOURNAL ANALYSIS

For the journals included in DBLP and ERA lists, we initially expected a similar influence of endogamy. However, after performing the same procedures we observed that the endogamy is not strongly influenced by the quality of the journal. In fact, with a study similar to that for the conference analysis, we obtained a maximum agreement of 62% (Figure 7). Although this number indicates some correlation, there is a big difference between the agreement of journals and conferences.

These results show that there is a behavior change in the way that people collaborate for publishing in journals, which can be explained in terms of previous studies. A recent survey among 22 editors from major software engineering journals report a general agreement that many publications in journals have archival intention and not innovative objectives [10]. Laender et al. [7] indicate that most journal papers have a conference prelude. Furthermore, the fraction of papers in journals that extend previous conference works has been estimated around 30% on average [4, 10], and for some journals it has been observed above 50% [10]. Since many works in journals focus on deeper analysis of previous ideas, journal publications benefit from groups of authors that have already collaborated. Therefore, we be-

lieve that the lower influence of endogamy in the case of journals is explained by a large set of journal papers from authors that collaborate again to extend ideas already presented in conference papers. For those journal papers, the endogamy approach is not indicative and alters the results.

7. CONCLUSIONS

The analysis introduced in this paper suggests that endogamy is a fundamental factor in understanding the generation of new scientific knowledge. The impact of social behavior in science is still a relatively unexplored topic, whose deeper understanding could be used to improve the efficiency in research innovation and effective team formation.

We observe that papers published in highly reputed conferences are published by groups of authors with low endogamy. On the other hand, low quality conferences tend to publish articles where authors have collaborated in many occasions. This stresses the importance of social contact in research and the opportunity that conferences offer to exchange new ideas and start collaborations.

We have also observed that high impact research in computer science does not have a unique strategy. Journal impact is not affected by endogamy in contrast to results in other research areas [6]. Although this seems a peculiar consequence of the extended versioning and archival focus of many computer science journals, we believe that it will be interesting to analyze the factors that determine the impact in computer science journal papers.

Our results show that endogamy could be used as a feature for determining the quality of conferences and, in particular, this applies to database conferences [9]. The endogamy of a group of authors can be computed when the paper is just published, in contrast to the number of citations to a paper, which may require years to be collected. Since an evaluation metric relying only on endogamy could be easily abused by dishonest conferences (by simply accepting papers that have small endogamy) we believe that endogamy should be taken as a complement to other metrics to obtain fast evaluation of conferences. An interesting research topic could be whether it is possible to design metrics based on endogamy which are difficult to flaw.

Acknowledgements

The authors thank the Ministry of Science and Innovation of Spain for grants TIN2009-14560-C03-03, PTQ-11-04970; and Generalitat de Catalunya for grant GRC-1087.

8. REFERENCES

[1] J. Bollen, H. Van de Sompel, A. Hagberg, and R. Chute. A principal component analysis of 39 scientific impact measures. *PloS one*, 4(6):e6022, 2009.

[2] K. Borner, L. Dall'Asta, W. Ke, and A. Vespignani. Studying the emerging global brain: Analyzing and visualizing the impact of co-authorship teams. *Complexity*, 10(4):57–67, 2005.

[3] E. Cortez, A. da Silva, and Gonçalves et al. FLUX-CIM: Flexible unsupervised extraction of citation metadata. In *Proc. JCDL*, pages 215–224, 2007.

[4] M. Eckmann, A. Rocha, and J. Wainer. Relationship between high-quality journals and conferences in computer vision. *Scientometrics*, 90(2):617–630, 2012.

[5] E. Garfield. Citation indexes for science: A new dimension in documentation through association of ideas. *Science*, 122(3159):108, 1955.

[6] R. Guimerà, B. Uzzi, J. Spiro, and L. Nunes. Team assembly mechanisms determine collaboration network structure and team performance. *Science*, 308:697–702, 2005.

[7] A. Laender, C. de Lucena, et al. Assessing the research and education quality of the top brazilian computer science graduate programs. *ACM SIGCSE Bulletin*, 40(2):135–145, 2008.

[8] S. Lopez-Montolio. Research endogamy as an indicator of conference quality. *UPC Master Thesis*, 2013.

[9] W. Martins, M. Gonçalves, et al. Learning to assess the quality of scientific conferences: a case study in computer science. In *Proc. JCDL*, pages 193–202, 2009.

[10] M. Montesi and J. Owen. From conference to journal publication: How conference papers in software engineering are extended for publication in journals. *J. Am. Soc. Inf. Sci. Technol.*, 59(5):816–829, 2008.

[11] M. Newman. Coauthorship networks and patterns of scientific collaboration. *Nat. Ac. Sc. USA*, 101(1):5200–5205, 2004.

[12] R. Porkess. *Statistics defined and explained*, page 64. Collins, 2005.

[13] E. Rahm and A. Thor. Citation analysis of database publications. *SIGMOD Record*, 34(4):48–53, 2005.

[14] Z. Zhuang, E. Elmacioglu, D. Lee, and C. Giles. Measuring conference quality by mining program committee characteristics. In *Proc. JCDL*, pages 225–234, 2007.

Information Diffusion in Online Social Networks: A Survey

Adrien Guille[1] Hakim Hacid[2] Cécile Favre[1] Djamel A. Zighed[1,3]
[1]ERIC Lab, Lyon 2 University, France
{firstname.lastname}@univ-lyon2.fr
[2]Bell Labs France, Alcatel-Lucent, France
hakim.hacid@alcatel-lucent.com
[3]Institute of Human Science, Lyon 2 University, France
abdelkader.zighed@ish-lyon.cnrs.fr

ABSTRACT

Online social networks play a major role in the spread of information at very large scale. A lot of effort have been made in order to understand this phenomenon, ranging from popular topic detection to information diffusion modeling, including influential spreaders identification. In this article, we present a survey of representative methods dealing with these issues and propose a taxonomy that summarizes the state-of-the-art. The objective is to provide a comprehensive analysis and guide of existing efforts around information diffusion in social networks. This survey is intended to help researchers in quickly understanding existing works and possible improvements to bring.

1. INTRODUCTION

Online social networks allow hundreds of millions of Internet users worldwide to produce and consume content. They provide access to a very vast source of information on an unprecedented scale. Online social networks play a major role in the diffusion of information by increasing the spread of novel information and diverse viewpoints [3]. They have proved to be very powerful in many situations, like Facebook during the 2010 Arab spring [22] or Twitter during the 2008 U.S. presidential elections [23] for instance. Given the impact of online social networks on society, the recent focus is on extracting valuable information from this huge amount of data. Events, issues, interests, *etc.* happen and evolve very quickly in social networks and their capture, understanding, visualization, and prediction are becoming critical expectations from both end-users and researchers. This is motivated by the fact that understanding the dynamics of these networks may help in better following events (*e.g.* analyzing revolutionary waves), solving issues (*e.g.* preventing terrorist attacks, anticipating natural hazards), optimizing business performance (*e.g.* optimizing social marketing campaigns), *etc.* Therefore researchers have in recent years developed a variety of techniques and models to capture information diffusion in online social networks, analyze it, extract knowledge from it and predict it.

Information diffusion is a vast research domain and has attracted research interests from many fields, such as physics, biology, *etc.* The diffusion of innovation over a network is one of the original reasons for studying networks and the spread of disease among a population has been studied for centuries. As computer scientists, we focus here on the particular case of information diffusion in online social networks, that raises the following questions : (i) *which pieces of information or topics are popular and diffuse the most*, (ii) *how, why and through which paths information is diffusing, and will be diffused in the future*, (iii) *which members of the network play important roles in the spreading process?*

The main goal of this paper is to review developments regarding these issues in order to provide a simplified view of the field. With this in mind, we point out strengths and weaknesses of existing approaches and structure them in a taxonomy. This study is designed to serve as guidelines for scientists and practitioners who intend to design new methods in this area. This also will be helpful for developers who intend to apply existing techniques on specific problems since we present a library of existing approaches in this area.

The rest of this paper is organized as follows. In Section 2 we detail online social networks basic characteristics and information diffusion properties. In Section 3 we present methods to detect topics of interest in social networks using information diffusion properties. Then we discuss how to model in-

formation diffusion and detail both explanatory and predictive models in Section 4. Next, we present methods to identify influential information spreaders in Section 5. In the last section we summarize the reviewed methods in a taxonomy, discuss their shortcomings and indicate open questions.

2. BASICS OF ONLINE SOCIAL NETWORKS AND INFORMATION DIFFUSION

An online social network (*OSN*) results from the use of a dedicated web-service, often referred to as *social network site* (*SNS*), that allows its users to (i) create a profile page and publish messages, and (ii) explicitly connect to other users thus creating social relationships. *De facto*, an OSN can be described as a user-generated content system that permits its users to communicate and share information.

An OSN is formally represented by a graph, where nodes are users and edges are relationships that can be either directed or not depending on how the SNS manages relationships. More precisely, it depends on whether it allows connecting in an unilateral (*e.g.* Twitter social model of *following*) or bilateral (*e.g.* Facebook social model of *friendship*) manner. Messages are the main information vehicle in such services. Users publish messages to share or forward various kinds of information, such as product recommendations, political opinions, ideas, *etc.* A message is described by (i) a text, (ii) an author, (iii) a time-stamp and optionally, (iv) the set of people (called "mentioned users" in the social networking jargon) to whom the message is specifically targeted. Figure 1 shows an OSN represented by a directed graph enriched by the messages published by its four members. An arc $e = (u_x, u_y)$ means that the user "u_x" is exposed to the messages published by "u_y". This representation reveals that, for example, the user named "u_1" is exposed to the content shared by "u_2" and "u_3". It also indicates that no one receives the messages written by "u_4".

DEFINITION 1 (Topic). *A coherent set of semantically related terms that express a single argument. In practice, we find three interpretations of this definition: (i) a set S of terms, with $|S| = 1$, e.g. {"obama"} (ii) a set S of terms, with $|S| > 1$, e.g. {"obama", "visit", "china"} and (iii) a probability distribution over a set S of terms.*

Every piece of information can be transformed into a topic [6, 30] using one of the common formalisms detailed in Definition 1. Globally, the content produced by the members of an OSN is a stream of messages. Figure 2 represents the stream produced by the members of the network depicted in the previous example. That stream can be viewed as a sequence of decisions (*i.e.* whether to adopt a certain topic or not), with later people watching the actions of earlier people. Therefore, individuals are influenced by the actions taken by others. This effect is known as *social influence* [2], and is defined as follows:

DEFINITION 2 (Social Influence). *A social phenomenon that individuals can undergo or exert, also called imitation, translating the fact that actions of a user can induce his connections to behave in a similar way. Influence appears explicitly when someone "retweets" someone else for example.*

DEFINITION 3 (Herd behavior). *A social behavior occurring when a sequence of individuals make an identical action, not necessarily ignoring their private information signals.*

DEFINITION 4 (Information Cascade). *A behavior of information adoption by people in a social network resulting from the fact that people ignore their own information signals and make decisions from inferences based on earlier people's actions.*

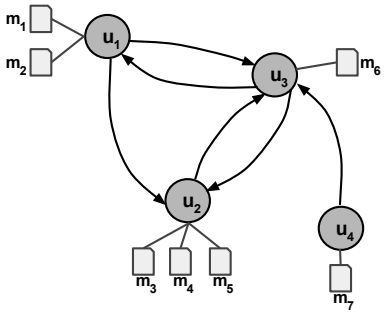


Figure 1: An example of OSN enriched by users' messages. Users are denoted u_i and messages m_j. An arc (u_x, u_y) means that u_x is exposed to the messages published by u_y.

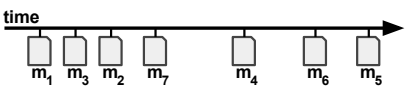


Figure 2: The stream of messages produced by the members of the network depicted on Figure 1.

Based on the social influence effect, information can spread across the network through the principles of *herd behavior* and *informational cascade* which we define respectively in Definition 3 and 4. In this context, some topics can become extremely popular, spread worldwide, and contribute to new trends. Eventually, the ingredients of an information diffusion process taking place in an OSN can be summarized as follows: (i) a piece of information carried by messages, (ii) spreads along the edges of the network according to particular mechanics, (iii) depending on specific properties of the edges and nodes. In the following sections, we will discuss these different aspects with the most relevant recent work related to them as well as an analysis of weaknesses, strength, and possible improvements for each aspect.

3. DETECTING POPULAR TOPICS

One of the main tasks when studying information diffusion is to develop automatic means to provide a global view of the topics that are popular over time or will become popular, and animate the network. This involves extracting "tables of content" to sum up discussions, recommending popular topics to users, or predicting future popular topics.

Traditional topic detection techniques developed to analyze static corpora are not adapted to message streams generated by OSNs. In order to efficiently detect topics in textual streams, it has been suggested to focus on bursts. In his seminal work, Kleinberg [26] proposes a state machine to model the arrival times of documents in a stream in order to identify bursts, assuming that all the documents belong to the same topic. Leskovec *et al.* [27] show that the temporal dynamics of the most popular topics in social media are indeed made up of a succession of rising and falling patterns of popularity, in other words, successive bursts of popularity. Figure 3 shows a typical example of the temporal dynamics of top topics in OSNs.

DEFINITION 5 (BURSTY TOPIC). *A behavior associated to a topic within a time interval in which it has been extensively treated but rarely before and after.*

In the following, we detail methods designed to detect topics that have drawn bursts of interest, *i.e. bursty topics* (see Definition 5), from a stream of topically diverse messages.

All approaches detailed hereafter rely on the computation of some frequencies and work on discrete data. Therefore they require the stream of messages to be discretized. This is done by transforming the raw continuous data into a sequence of collection of messages published during equally sized time slices. This principle is illustrated on Figure 4, which shows a possible discretization of the stream previously depicted in Figure 2. This pre-processing step is not trivial since it defines the granularity of the topic detection. A very fine discretization (*i.e.* short time-slices) will allow to detect topics that were popular during short periods whereas a discretization using longer time-slices will not.

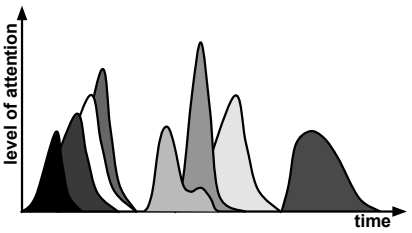


Figure 3: Temporal dynamics of popular topics. Each shade of gray represents a topic.

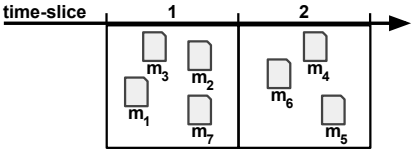


Figure 4: A possible discretization of the stream of messages shown on Figure 2.

Shamma *et al.* [46] propose a simple model, *PT* (*i.e. Peaky Topics*) , similar to the classical tf-idf model [44] in the sense that it is based on a normalized term frequency metric. In order to quantify the overall term usage, they consider each time slice as a pseudo-document composed of all the messages in the corresponding collection. The normalized term frequency ntf is defined as follows: $ntf_{t,i} = \frac{tf_{t,i}}{cf_t}$, where $tf_{t,i}$ is the frequency of term t at the i^{th} time slice and cf_t is the frequency of term t in the whole message stream. Using that metric, bursty topics defined as single terms are ranked. However, some terms can be polysemous or ambiguous and a single term doesn't seem to be enough to clearly identify a topic. Therefore, more sophisticated methods have been developed.

AlSumait *et al.* [1] propose an online topic model, more precisely, a non-Markov on-line LDA Gibbs sampler topic model, called *OLDA*. Basically, LDA (*i.e.* Latent Dirichlet Allocation [4]) is a statistical generative model that relies on a hierarchical Bayesian network that relates words and mes-

sages through latent topics. The generative process behind is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. The idea of *OLDA* is to incrementally update the topic model at each time slice using the previously generated model as a prior and the corresponding collection of messages to guide the learning of the new generative process. This method builds an evolutionary matrix for each topic that captures the evolution of the topic over time and thus permits to detect bursty topics.

Cataldi *et al.* [6] propose the *TSTE* method (*i.e.* Temporal and Social Terms Evaluation) that considers both temporal and social properties of the stream of messages. To this end, they develop a five-step process that firstly formalize the messages content as vectors of terms with their relative frequencies computed by using the augmented normalized term frequency [43]. Then, the authority of the active authors is assessed using their relationships and the Page Rank algorithm [35]. It allows to model the life cycle of each term on the basis of a biological metaphor, which is based on the calculation of values of nutrition and energy that leverage the users authority. Using supervised or unsupervised techniques, rooted in the calculation of a critical drop value based on the energy, the proposed method can identify most bursty terms. Finally, a solution is provided to define bursty topics as sets of terms using a co-occurence based metric.

These methods identify particular topics that have drawn bursts of interest in the past. Lu *et al.* [40] develop a method that permits predicting which topics will draw attention in the near future. Authors propose to adapt a technical analysis indicator primary used for stock price study, namely *MACD* (*i.e.* Moving Average Convergence Divergence), to identify bursty topics, defined as a single term. The principle of *MACD* is to turn two trend-following indicators, precisely a short period and a longer period moving average of terms frequency, into a momentum oscillator. The trend momentum is obtained by calculating the difference between the long and the shorter moving averages. Authors give two simple rules to identify when the trends of a term will rise: (i) when the value of the trend momentum changes from negative to positive, the topic is beginning to rise; (ii) when the value changes from positive to negative, the level of attention given to the topic is falling.

The above methods are based on the detection of unusual term frequencies in exchanged messages to detect interesting topics in OSNs. However, more and more frequently, OSNs users publish non-textual content such as URL, pictures or videos. To deal with non-textual content, Takahashi *et al.* [47] propose to use mentions contained in messages to identify bursty topics, instead of focusing on the textual content. Mentioning is a social practice used to explicitly target messages and eventually engage discussion. For that, they develop a method that combines a *mentioning anomaly score* and a change-point detection technique based on *SDNML* (*i.e.* Sequentially Discounting Normalized Maximum Likelihood). The anomaly is calculated with respect to the standard mentioning behavior of each user, which is estimated by a probability model.

Table 1 summarizes the surveyed methods according to four axes. The table is structured according to four main criteria that allow for a quick comparison: (i) how is a topic defined, (ii) which dimensions are incorporated into each method, (iii) which types of content each method can handle, and (iv) either the method detects actual bursts or predicts them. It should be noted that the table is not intended to express any preference regarding one method or another, but rather to present a global comparison.

reference	**topic definition**			**dimension(s)**		**content type**		**task type**	
	single term	set of terms	distribution	content	social	textual	non-textual	observation	prediction
PT	x			x		x		x	
OLDA			x	x		x		x	
TSTE		x		x	x	x		x	
SDNML	x				x	x	x	x	
MACD	x			x		x			x

Table 1: Summary of topic detection approaches w.r.t topic definition, incorporated dimensions, handled content and the task.

4. MODELING INFORMATION DIFFUSION

Modeling how information spreads is of outstanding interest for stopping the spread of viruses, analyzing how misinformation spread, *etc.* In this section, we first give the basics of diffusion modeling

and then detail the different models proposed to capture or predict spreading processes in OSNs.

DEFINITION 6 (Activation Sequence). *An ordered set of nodes capturing the order in which the nodes of the network adopted a piece of information.*

DEFINITION 7 (Spreading Cascade). *A directed tree having as a root the first node of the activation sequence. The tree captures the influence between nodes (branches represent who transmitted the information to whom) and unfolds in the same order as the activation sequence.*

The diffusion process is characterized by two aspects: its structure, *i.e.* the diffusion graph that transcribes who influenced whom, and its temporal dynamics, *i.e.* the evolution of the diffusion rate which is defined as the amount of nodes that adopts the piece of information over time. The simplest way to describe the spreading process is to consider that a node can be either activated (*i.e.* has received the information and tries to propagate it) or not. Thus, the propagation process can be viewed as a successive activation of nodes throughout the network, called *activation sequence*, defined in Definition 6.

Usually, models developed in the context of OSNs assume that people are only influenced by actions taken by their connections. To put it differently, they consider that an OSN is a closed world and assume that information spreads because of informational cascades. That is why the path followed by a piece of information in the network (*i.e.* the diffusion graph) is often referred to as the *spreading cascade*, defined in Definition 7. Activation sequences are simply extracted from data by collecting messages dealing with the studied information, *i.e.* topic, and ordering them according to the time axis. This principle is illustrated in Figure 5. It provides knowledge about where and when a piece of information propagated but not how and why did it propagate. Therefore, there is a need for models that can capture and predict the hidden mechanism underlying diffusion. We can distinguish two categories of models in this scope: (i) explanatory models and (ii) predictive models. In the following, we detail these two categories and analyze some representative efforts in both of them.

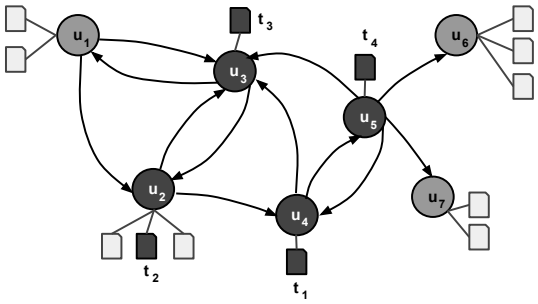


Figure 5: An OSN in which darker nodes took part in the diffusion process of a particular information. The activation sequence can be extracted using the time at which the messages were published: $[u_4; u_2; u_3; u_5]$, with $t_1 < t_2 < t_3 < t_4$.

4.1 Explanatory Models

The aim of explanatory models is to infer the underlying spreading cascade, given a complete activation sequence. These models make it possible to retrace the path taken by a piece of information and are very useful to understand how information propagated.

Gomez *et al.* [15] propose to explore correlations in nodes infections times to infer the structure of the spreading cascade and assume that activated nodes influence each of their neighbors independently with some probability. Thus, the probability that one node had transmitted information to another is decreasing in the difference of their activation time. They develop *NETINF*, an iterative algorithm based on submodular function optimization for finding the spreading cascade that maximizes the likelihood of observed data.

Gomez *et al.* [14] extend *NETINF* and propose to model the diffusion process as a spatially discrete network of continuous, conditionally independent temporal processes occurring at different rates. The likelihood of a node infecting another at a given time is modeled via a probability density function depending on infection times and the transmission rate between the two nodes. The proposed algorithm, *NETRATE*, infers pairwise transmission rates and the graph of diffusion by formulating and solving a convex maximum likelihood problem [9].

These methods consider that the underlying network remains static over time. This is not a satisfying assumption, since the topology of OSNs evolves very quickly, both in terms of edges creation and deletion. For that reason, Gomez *et al.* [16] extend *NETRATE* and propose a time-varying inference algorithm, *INFOPATH*, that uses stochastic gradients to provide on-line estimates of the structure and temporal dynamics of a network that changes over time.

In addition, because of technical and crawling API limitations, there is a *data acquisition bottle-*

reference	network		inferred properties			supports missing data
	static	dynamic	pairwaise transmission probability	pairwaise transmission rate	cascade properties	
NETINF	x		x		x	
NETRATE	x		x	x	x	
INFOPATH	x	x	x	x	x	
k-tree model	x				x	x

Table 2: Summary of explanatory models w.r.t the nature of the underlying network, inferred properties and the ability of the method to work with incomplete data.

neck potentially responsible for missing data. To overcome this issue, one approach is to crawl data as efficiently as possible. Choudhury *et al.* [7] analysed how the data sampling strategy impacts the discovery of information diffusion in social media. Based on experimentations on Twitter data, they concluded that sampling methods that consider both network topology and users' attributes such as activity and localisation allow to capture information diffusion with lower error in comparison to naive strategies, like random or activity-only based sampling. Another approach is to develop specific models that assume that data are missing. Sadikov *et al.* [41] develop a method based on a *k*-tree model designed to, given only a fraction of the complete activation sequence, estimate the properties of the complete spreading cascade, such as its size or depth.

We summarize the surveyed explanatory models in Table 2. In the following, we detail the second category of models, namely, predictive models.

4.2 Predictive Models

These models aim at predicting how a specific diffusion process would unfold in a given network, from temporal and/or spatial points of view by learning from past diffusion traces. We classify existing models into two development axes, graph and non-graph based approaches.

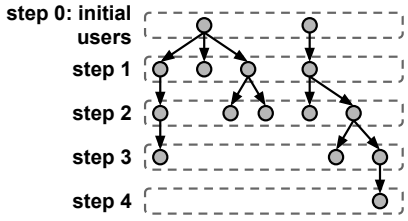


Figure 6: A spreading process modeled by Independent Cascades in four steps.

4.2.1 Graph based approaches

There are two seminal models in this category, namely *Independent Cascades* (*IC*) [13] and *Linear Threshold* (*LT*) [17]. They assume the existence of a static graph structure underlying the diffusion and focus on the structure of the process. They are based on a directed graph where each node can be activated or not with a monotonicity assumption, i.e. activated nodes cannot deactivate. The *IC* model requires a diffusion probability to be associated to each edge whereas *LT* requires an influence degree to be defined on each edge and an influence threshold for each node. For both models, the diffusion process proceeds iteratively in a synchronous way along a discrete time-axis, starting from a set of initially activated nodes, commonly named *early adopters* [37]:

DEFINITION 8 (Early Adopters). *A set of users who are the first to adopt a piece of information and then trigger its diffusion.*

In the case of *IC*, for each iteration, the newly activated nodes try once to activate their neighbors with the probability defined on the edge joining them. In the case of *LT*, at each iteration, the inactive nodes are activated by their activated neighbors if the sum of influence degrees exceeds their own influence threshold. Successful activations are effective at the next iteration. In both cases, the process ends when no new transmission is possible, *i.e.* no neighboring node can be contacted. These two mechanisms reflect two different points of view: *IC* is sender-centric while *LT* is receiver-centric. An example of spreading process modeled with *IC* is given by Figure 6. We detail hereafter models arising from those approaches and adapted to OSNs.

Galuba *et al.* [11] propose to use the *LT* model to predict the graph of diffusion, having already observed the beginning of the process. Their model relies on parameters such as information virality, pairwise users degree of influence and user probability of adopting any information. The *LT* model

is fitted on the data describing the beginning of the diffusion process by optimizing the parameters using the gradient ascent method. However, LT can't reproduce realistic temporal dynamics.

Saito *et al.* [42] relax the synchronicity assumption of traditional IC and LT graph-based models by proposing asynchronous extensions. Named $AsIC$ and $AsLT$ (*i.e.* asynchronous independent cascades and asynchronous linear threshold), they proceed iteratively along a continuous time axis and require the same parameters as their synchronous counterparts plus a time-delay parameter on each edge of the graph. Models parameters are defined in a parametric way and authors provide a method to learn the functional dependency of the model parameters from nodes attributes. They formulate the task as a maximum likelihood estimation problem and an update algorithm that guarantees the convergence is derived. However, they only experimented with synthetic data and don't provide a practical solution.

Guille *et al.* [19] also model the propagation process as asynchronous independent cascades. They develop the *T-BaSIC* model (*i.e.* Time-Based Asynchronous Independent Cascades), which parameters aren't fixed numerical values but functions depending on time. The model parameters are estimated from social, semantic and temporal nodes' features using logistic regression.

4.2.2 Non-graph based approaches

Non-graph based approaches do not assume the existence of a specific graph structure and have been mainly developed to model epidemiological processes. They classify nodes into several classes (*i.e.* states) and focus on the evolution of the proportions of nodes in each class. *SIR* and *SIS* are the two seminal models [21, 34], where S stands for "susceptible", I for "infected" (*i.e.* adopted the information) and R for recovered (*i.e.* refractory). In both cases, nodes in the S class switch to the I class with a fixed probability β. Then, in the case of SIS, nodes in the I class switch to the S class with a fixed probability γ, whereas in the case of SIR they permanently switch to the R class. The percentage of nodes in each class is expressed by simple differential equations. Both models assume that every node has the same probability to be connected to another and thus connections inside the population are made at random.

Leskovec *et al.* [28] propose a simple and intuitive SIS model that requires a single parameter, β. It assumes that all nodes have the same probability β to adopt the information and nodes that have adopted the information become susceptible at the next time-step (*i.e.* $\gamma = 1$). This is a strong assumption since in real-world social networks, influence is not evenly distributed between all nodes and it is necessary to develop more complex modeling that take into account this characteristic.

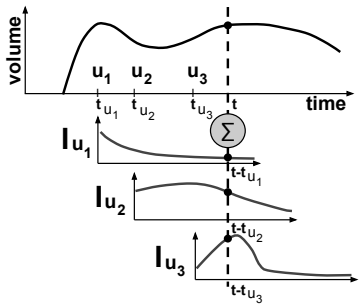


Figure 7: LIM forecasts the rate of diffusion by summing the influence functions of a given set of early adopters. Here, the early adopters are u_1, u_2 and u_3 whose respective influence functions are Iu_1, Iu_2 and Iu_3.

Yang *et al.* [50] start from the assumption that the diffusion of information is governed by the influence of individual nodes. The method focuses on predicting the temporal dynamics of information diffusion, under the form of a time-series describing the rate of diffusion of a piece of information, *i.e.* the volume of nodes that adopt the information through time. They develop a Linear Influence model (LIM), where the influence functions of individual nodes govern the overall rate of diffusion. The influence functions are represented in a non-parametric way and are estimated by solving a non-negative least squares problem using the Reflective Newton Method [8]. Figure 7 illustrates how LIM forecasts the rate of diffusion from a set of early adopters and their activation time.

Wang *et al.* [48] propose a Partial Differential Equation (PDE) based model to predict the diffusion of an information injected in the network by a given node. More precisely, a diffusive logistic equation model is used to predict both topological and temporal dynamics. Here, the topology of the network is considered only in term of the distance from each node to the source node. The dynamics of the process is given by a logistic equation that models the density of influenced users at a given distance of the source and at a given time. That definition of the network topology allows to formulate the problem simply, as for classical non-graph based methods while integrating some spatial knowledge. The

reference	dimension(s)			basis		mathematical modeling	
	social	time	content	graph based	non-graph based	parametric	non-parametric
LT-based	x		x	x		x	
AsIC, AsLT	n/a	n/a	n/a	x		x	
T-BaSIC	x	x	x	x		x	
SIS-based		x			x	x	
LIM	x	x			x		x
PDE	x	x			x	x	

Table 3: Summary of diffusion prediction methods, distinguishing graph and non-graph based approaches w.r.t incorporated dimensions and mathematical modeling.

parameters of the model are estimated using the Cubic Spline Interpolation method [12].

We summarize the surveyed predictive models in Table 3. In the following section, we discuss the role of nodes in the propagation process and how to identify influential spreaders.

5. IDENTIFYING INFLUENTIAL INFORMATION SPREADERS

Identifying the most influential spreaders in a network is critical for ensuring efficient diffusion of information. For instance, a social media campaign can be optimized by targeting influential individuals who can trigger large cascades of further adoptions. This section presents briefly some methods that illustrate the various possible ways to measure the relative importance and influence of each node in an online social network.

DEFINITION 9 (K-Core). *Let G be a graph. If H is a sub-graph of G, $\sigma(H)$ will denote the minimum degree of H. Thus each node of H is adjacent to at least $\sigma(H)$ other nodes of H. If H is a maximal connected (induced) sub-graph of G with $\sigma(H) >= k$, we say that H is a k-core of G [45].*

Kitsak *et al.* [25] show that the best spreaders are not necessarily the most connected people in the network. They find that the most efficient spreaders are those located within the *core* of the network as identified by the k-core decomposition analysis [45], as defined in Definition 9. Basically, the principle of the k-core decomposition is to assign a core index k_s to each node such that nodes with the lowest values are located at the periphery of the network while nodes with the highest values are located in the center of the network. The innermost nodes thus forms the core of the network. Brown *et al.* [5] observe that the results of the k-shell decomposition on Twitter network are highly skewed. Therefore they propose a modified algorithm that uses a logarithmic mapping, in order to produce fewer and more meaningful k-shell values.

Cataldi *et al.* [6] propose to use the well known *PageRank* algorithm [35] to assess the distribution of influence throughout the network. The *PageRank* value of a given node is proportional to the probability of visiting that node in a random walk of the social network, where the set of states of the random walk is the set of nodes.

The methods we have just described only exploit the topology of the network, and ignore other important properties, such as nodes' features and the way they process information. Starting from the observation that most OSNs members are passive information consumers, Romero *et al.* [38] develop a graph-based approach similar to the well known *HITS* algorithm, *IP* (*i.e. Influence-Passivity*), that assigns a relative influence and a passivity score to every users based on the ratio at which they forward information. However, no individual can be a universal influencer, and influential members of the network tend to be influential only in one or some specific domains of knowledge. Therefore, Pal *et al.* [36] develop a non-graph based, topic-sensitive method. To do so, they define a set of nodal and topical features for characterizing the network members. Using probabilistic clustering over this feature space, they rank nodes with a within-cluster ranking procedure to identify the most influential and authoritative people for a given topic. Weng *et al.* [49] also develop a topic-sensitive version of the Page Rank algorithm dedicated to Twitter, TwitterRank.

Kempe *et al.* [24] adopt a different approach and propose to use the *IC* and *LT* models (previously described in Section 4.2.1) to tackle the influence maximization problem. This problem asks, for a parameter k, to find a k-node set of maximum influence in the network. The influence of a given set of nodes corresponds to the number of activated nodes at the end of the diffusion process according

reference	graph based	incorporated dimension(s)	
		users' features	topic
k-shell decomposition	x		
log k-shell decomposition	x		
PageRank	x		
Topic-sensitive PageRank	x		x
IP	x	x	
Topical Authorities		x	x
k-node set	x		

Table 4: Summary of influential spreaders identification methods distinguishing graph and non-graph based approaches w.r.t incorporated dimensions.

to *IC* or *LT*, using this set as the set of initially activated nodes. They provide an approximation for this optimization problem using a greedy hill-climbing strategy based on submodular functions.

The surveyed influence assessment methods are summarized in Table 4.

6. DISCUSSION

In this article, we surveyed representative and state-of-the-art methods related to information diffusion analysis in online social networks, ranging from popular topic detection to diffusion modeling techniques, including methods for identifying influential spreaders. Figure 8 presents the taxonomy of the various approaches employed to address these issues. Hereafter we provide a discussion regarding their shortcomings and related open problems.

6.1 Detecting Popular Topics

The detection of popular topics from the stream of messages produced by the members of an OSN relies on the identification of *bursts*. There are mainly two ways to detect such patterns, by analyzing (i) term frequency or (ii) social interaction frequency. In this area, the following challenges certainly need to be addressed:

Topic definition and scalability. It is obvious that not all methods define a topic in the same way. For instance *Peaky Topics* simply assimilates a topic to a word. It has the advantage to be a low complexity solution, however, the produced result is of little interest. In contrast, *OLDA* defines a topic as a distribution over a set of words but in turn has a high complexity, which prevents it from being applied at large scale. Consequently, there is a need for new methods that could produce intelligible results while preserving efficiency. We identify two possible ways to do so, through: (i) the conception of new scalable algorithms, or (ii) improved implementations of the algorithms using, *e.g.* distributed systems (such as Hadoop).

Social dimension. Furthermore, popular topic detection could be improved by leveraging burstiness and people authority, as does *TSTE*, which relies on the *PageRank* algorithm. However, that possibility remains ill explored so far.

Data complexity. Currently the focus is set on the textual content exchanged in social networks. However, more and more often, users exchange other types of data such as images, videos, URLs pointing to those objects or Web pages, *etc.* This situation has to be fully considered and integrated at the heart of the efforts carried out to provide a complete solution for topic detection.

6.2 Modeling Information Diffusion

We distinguish two types of models, explanatory and predictive. Concerning predictive models, on the one hand there are non-graph based methods, that are limited by the fact that they ignore the topology of the network and only forecast the evolution of the rate at which information globally diffuses. On the other hand, there are graph based approaches that are able to predict who will influence whom. However, they cannot be used when the network is unknown or implicit. Although a lot of effort have been performed in this area, generally speaking, there is a need to consider more realistic constraints when studying information diffusion. In particular, the following issues have to be dealt with:

DEFINITION 10 (Closed World). *The closed world assumption holds that information can only propagate from node to node via the network edges and that nodes cannot be influenced by external sources.*

Closed world assumption. The major observation about modeling information diffusion is certainly that all the described approaches work under a closed world assumption, defined in Definition 10. In other words, they assume that people can only be influenced by other members of the network and that information spreads because of informational cascades. However, most observed spreading pro-

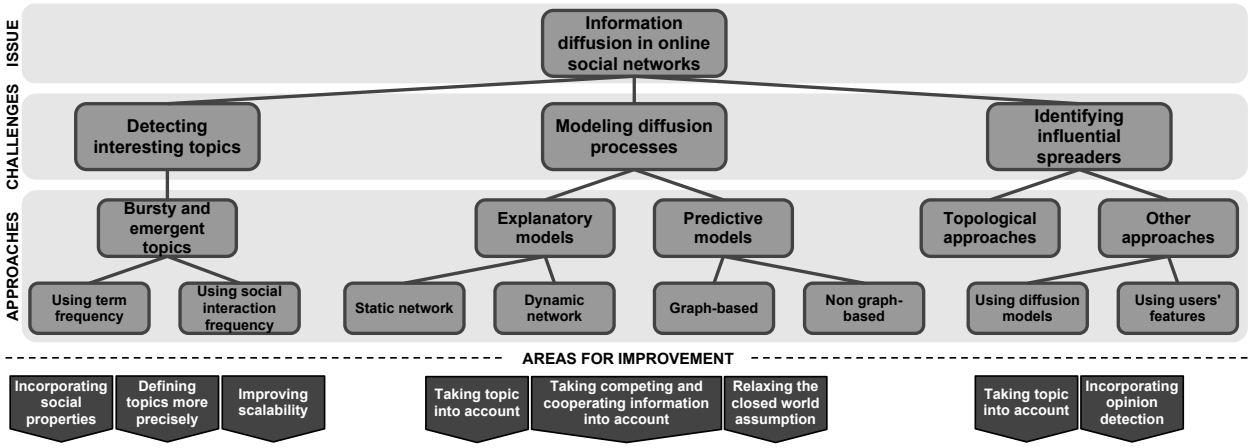


Figure 8: The above taxonomy presents the three main research challenges arising from information diffusion in online social networks and the related types of approaches, annotated with areas for improvement.

cesses in OSNs do not rely solely on social influence. The closed-world assumption is proven incorrect in recent work on Twitter done by Myers *et al.* [32] in which authors observe that information tends to jump across the network. The study shows that only 71% of the information volume in Twitter is due to internal influence and the remaining 29% can be attributed to external events and influence. Consequently they provide a model capable of quantifying the level of external exposure and influence using hazard functions [10]. To relax this assumption, one way would be to align users' profiles across multiple social networking sites. In this way, it would be possible to observe the information diffusion among various platforms simultaneously (subject to the availability of data). Some work tend to address this type of problems by proposing to de-anonymize the social networks [33].

Cooperating and competing diffusion processes. In addition, the described studies rely on the assumption that diffusion processes are independent, *i.e.* each information spreads in isolation. Myers *et al.* [31] argue that spreading processes cooperate and compete. Competing contagions decrease each other's probability of diffusion, while cooperating ones help each other in being adopted. They propose a model that quantifies how different spreading cascades interact with each other. It predicts diffusion probabilities that are on average 71% more or less than the diffusion probability would be for a purely independent diffusion process. We believe that models have to consider and incorporate this knowledge.

Topic-sensitive modeling. Furthermore, it is important for predictive models to be topic-sensitive. Romero *et al.* [39] have studied Twitter and found significant differences in the mechanics of information diffusion across topics. More particularly, they have observed that information dealing with politically controversial topics are particularly persistent, with repeated exposures continuing to have unusually large marginal effects on adoption, which validates the *complex contagion principle* that stipulates that repeated exposures to an idea are particularly crucial when the idea is controversial or contentious.

Dynamic networks. Finally, it is important to note that OSNs are highly dynamic structures. Nonetheless most of the existing work rely on the assumption that the network remains static over time. Integrating link prediction could be a basis to improve prediction accuracy. A more complete review of literature on this topic can be found in [20].

6.3 Identifying Influential Spreaders

There are various ways to tackle this issue, ranging from pure topological approaches, such as *k-shell decomposition* or *HITS* to textual clustering based approaches, including hybrid methods, such as *IP* which combines the *HITS* algorithm with nodes' features. As mentioned previously, there is no such thing as a universal influencer and therefore topic-sensitive methods have also been developed.

Opinion detection. The notion of influence is strongly linked to the notion of opinion. Numerous studies on this issue have emerged in recent years, aiming at automatically detecting opinions or sentiment from corpus of data. We believe that

it might be interesting to include this kind of work in the context of information diffusion. Work dealing with the diffusion of opinions themselves have emerged [29] and it seems that there is an interest to couple these approaches.

6.4 Applications

Even if there are a lot of contributions in the domain of online social networks dynamics analysis, we can remark that implementations are rarely provided for re-use. What is more, available implementations require different formatting of the input data and are written using various programming languages, which makes it hard to evaluate or compare existing techniques. *SONDY* [18] intends to facilitate the implementation and distribution of techniques for online social networks data mining. It is an open-source tool that provides data preprocessing functionalities and implements some of the methods reviewed in this paper for topic detection and influential spreaders identification. It features a user-friendly interface and proposes visualizations for topic trends and network structure.

7. REFERENCES

[1] L. AlSumait, D. Barbará, and C. Domeniconi. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *ICDM '08*, pages 3–12, 2008.

[2] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *KDD '08*, pages 7–15, 2008.

[3] E. Bakshy, I. Rosenn, C. Marlow, and L. Adamic. The role of social networks in information diffusion. In *WWW '12*, pages 519–528, 2012.

[4] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[5] P. Brown and J. Feng. Measuring user influence on Twitter using modified k-shell decomposition. In *ICWSM '11 Workshops*, pages 18–23, 2011.

[6] M. Cataldi, L. Di Caro, and C. Schifanella. Emerging topic detection on Twitter based on temporal and social terms evaluation. In *MDMKDD '10*, pages 4–13, 2010.

[7] M. D. Choudhury, Y.-R. Lin, H. Sundaram, K. S. Candan, L. Xie, and A. Kelliher. How does the data sampling strategy impact the discovery of information diffusion in social media? In *ICWSM '10*, pages 34–41, 2010.

[8] T. F. Coleman and Y. Li. A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM J. on Optimization*, 6(4):1040–1058, Apr. 1996.

[9] I. CVX Research. CVX: Matlab software for disciplined convex programming, version 2.0 beta. `http://cvxr.com/cvx`, sep 2012.

[10] R. C. Elandt-Johnson and N. L. Johnson. *Survival Models and Data Analysis*. John Wiley and Sons, 1980/1999.

[11] W. Galuba, K. Aberer, D. Chakraborty, Z. Despotovic, and W. Kellerer. Outtweeting the twitterers - predicting information cascades in microblogs. In *WOSN '10*, pages 3–11, 2010.

[12] C. F. Gerald and P. O. Wheatley. *Applied numerical analysis with MAPLE; 7th ed.* Addison-Wesley, Reading, MA, 2004.

[13] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 2001.

[14] M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *ICML '11*, pages 561–568, 2011.

[15] M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *KDD '10*, pages 1019–1028, 2010.

[16] M. Gomez-Rodriguez, J. Leskovec, and B. Schökopf. Structure and dynamics of information pathways in online media. In *WSDM '13*, pages 23–32, 2013.

[17] M. Granovetter. Threshold models of collective behavior. *American journal of sociology*, pages 1420–1443, 1978.

[18] A. Guille, C. Favre, H. Hacid, and D. Zighed. Sondy: An open source platform for social dynamics mining and analysis. In *SIGMOD '13*, (demonstration) 2013.

[19] A. Guille and H. Hacid. A predictive model for the temporal dynamics of information diffusion in online social networks. In *WWW '12 Companion*, pages 1145–1152, 2012.

[20] M. A. Hasan and M. J. Zaki. A survey of link prediction in social networks. In *Social Network Data Analytics*, pages 243–275. Springer, 2011.

[21] H. W. Hethcote. The mathematics of infectious diseases. *SIAM REVIEW*, 42(4):599–653, 2000.

[22] P. N. Howard and A. Duffy. Opening closed

regimes, what was the role of social media during the arab spring? *Project on Information Technology and Political Islam*, pages 1–30, 2011.

[23] A. Hughes and L. Palen. Twitter adoption and use in mass convergence and emergency events. *International Journal of Emergency Management*, 6(3):248–260, 2009.

[24] D. Kempe. Maximizing the spread of influence through a social network. In *KDD '03*, pages 137–146, 2003.

[25] M. Kitsak, L. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. Stanley, and H. Makse. Identification of influential spreaders in complex networks. *Nature Physics*, 6(11):888–893, Aug 2010.

[26] J. Kleinberg. Bursty and hierarchical structure in streams. In *KDD '02*, pages 91–101, 2002.

[27] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD '09*, pages 497–506, 2009.

[28] J. Leskovec, M. Mcglohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs. In *SDM '07*, pages 551–556, (short paper) 2007.

[29] L. Li, A. Scaglione, A. Swami, and Q. Zhao. Phase transition in opinion diffusion in social networks. In *ICASSP '12*, pages 3073–3076, 2012.

[30] J. Makkonen, H. Ahonen-Myka, and M. Salmenkivi. Simple semantics in topic detection and tracking. *Inf. Retr.*, 7(3-4):347–368, Sept. 2004.

[31] S. Myers and J. Leskovec. Clash of the contagions: Cooperation and competition in information diffusion. In *ICDM '12*, pages 539–548, 2012.

[32] S. A. Myers, C. Zhu, and J. Leskovec. Information diffusion and external influence in networks. In *KDD '12*, pages 33–41, 2012.

[33] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *SP '09*, pages 173–187, 2009.

[34] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[35] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *WWW '98*, pages 161–172, 1998.

[36] A. Pal and S. Counts. Identifying topical authorities in microblogs. In *WSDM '11*, pages 45–54, 2011.

[37] E. M. Rogers. *Diffusion of Innovations, 5th Edition*. Free Press, 5th edition, aug 2003.

[38] D. Romero, W. Galuba, S. Asur, and B. Huberman. Influence and passivity in social media. In *ECML/PKDD '11*, pages 18–33, 2011.

[39] D. M. Romero, B. Meeder, and J. Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on Twitter. In *WWW '11*, pages 695–704, 2011.

[40] L. Rong and Y. Qing. Trends analysis of news topics on Twitter. *International Journal of Machine Learning and Computing*, 2(3):327–332, 2012.

[41] E. Sadikov, M. Medina, J. Leskovec, and H. Garcia-Molina. Correcting for missing data in information cascades. In *WSDM '11*, pages 55–64, 2011.

[42] K. Saito, K. Ohara, Y. Yamagishi, M. Kimura, and H. Motoda. Learning diffusion probability based on node attributes in social networks. In *ISMIS '11*, pages 153–162, 2011.

[43] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, 1988.

[44] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1986.

[45] S. B. Seidman. Network structure and minimum degree. *Social Networks*, 5(3):269 – 287, 1983.

[46] D. A. Shamma, L. Kennedy, and E. F. Churchill. Peaks and persistence: modeling the shape of microblog conversations. In *CSCW '11*, pages 355–358, (short paper) 2011.

[47] T. Takahashi, R. Tomioka, and K. Yamanishi. Discovering emerging topics in social streams via link anomaly detection. In *ICDM '11*, pages 1230–1235, 2011.

[48] F. Wang, H. Wang, and K. Xu. Diffusive logistic model towards predicting information diffusion in online social networks. In *ICDCS '12 Workshops*, pages 133–139, 2012.

[49] J. Weng, E.-P. Lim, J. Jiang, and Q. He. TwitterRank: finding topic-sensitive influential twitterers. In *WSDM '10*, pages 261–270, 2010.

[50] J. Yang and J. Leskovec. Modeling information diffusion in implicit networks. In *ICDM '10*, pages 599–608, 2010.

Discovering Semantic Relations from the Web and Organizing them with PATTY

Ndapandula Nakashole, Gerhard Weikum, Fabian Suchanek
Max Planck Institute for Informatics, Saarbruecken, Germany
{nnakasho,weikum,suchanek}@mpi-inf.mpg.de

ABSTRACT

PATTY is a system for automatically distilling relational patterns from the Web, for example, the pattern "X covered Y" between a singer and someone else's song. We have extracted a large collection of such patterns and organized them in a taxonomic manner, similar in style to the WordNet thesaurus but capturing relations (binary predicates) instead of concepts and classes (unary predicates). The patterns are organized by semantic types and synonyms, and they form a hierarchy based on subsumptions. For example, "X covered Y" is subsumed by "X sang Y", which in turn is subsumed by "X performed Y" (where X can be any musician, not just a singer).
In this paper we give an overview of the PATTY system and the resulting collections of relational patterns. We discuss the four main components of PATTY's architecture and a variety of use cases, including the paraphrasing of relations, and semantic search over subject-predicate-object triples. This kind of search can handle entities, relations, semantic types, noun phrases, and relational phrases.

1. INTRODUCTION

Ongoing efforts to extract information from Web data have produced large-scale knowledge bases (KBs) [1, 2, 3, 13]. These KBs store information about real-world entities, such as people, cities, or movies. The KBs mostly use the RDF triple format to store the data. Each triple contains a subject, a predicate, and an object. For example, the fact that Amy Winehouse was born in South Gate would be stored as the triple ⟨*Amy_Winehouse, wasBornIn, South_Gate*⟩. The predicates of such triples are called *relations*. Most KBs contain a limited number of "standard" relations such as *wasBornIn* and *isMarriedTo*. However, there are many more relations that are often missing. For example, in the music domain, one might be interested in relations such as *sang*, *coveredSong* and *hadDuetWith*. Before even populating such relations with triples, one has to find which relations exist. With the PATTY project [10, 11, 12], we embarked on automatically mining new relations from the Web.

Mining relations from the Web is difficult, because relationships between entities are expressed in highly diverse and noisy forms in natural-language text. For example, Web sources may use the verbal phrases ⟨*X's voice in Y*⟩ or ⟨*X's performance of the song Y*⟩ to say that a person sang a song. We call these verbal phrases *patterns*, as opposed to the canonical relation *sang*. So the same relation can be expressed with different patterns. Conversely, the same pattern may denote different relations. For example, ⟨*X covered Y*⟩ could refer to a singer performing someone else's song or to a book covering a historic event (e.g., "War_and_Peace covered Napoleonic_Wars").

Understanding the semantic equivalence of patterns and mapping them to canonical relations is the core challenge in relational information extraction (IE). This problem arises both in seed-based distantly supervised IE with explicitly specified target relations, and in Open IE where the relations themselves are unknown a priori and need to be discovered in an unsupervised manner. Comprehensively gathering and systematically organizing patterns for an *open set of relations* is the problem addressed by the PATTY system.

The approach we take in PATTY is to systematically harvest textual patterns from text corpora. We group synonymous patterns into pattern synsets, so that patterns that express the same relationship are grouped together. We organize these synsets into a subsumption hierarchy, where more general relationships (such as *performed*) subsume more special relationships (such as *sang*). PATTY makes use of a generalized notion of *ontologically typed patterns*. These patterns have a type signature for the entities that they connect, as in ⟨⟨*person*⟩ *sang* ⟨*song*⟩⟩. The type signatures are derived through the use of a dictionary of entity-class pairs, provided by knowledge bases like YAGO[13], Freebase [2], or DBpedia[1].

This paper gives an overview of PATTY based on work reported in [10], [11], and [12]. We first present the design of the main components of PATTY's architecture: the pattern extraction, the SOL pattern model, the pattern

generalization, and the subsumption mining. We then present various applications that can make use of the PATTY data.

The PATTY collections of relational phrases are freely available at the URL `http://www.mpi-inf.mpg.de/yago-naga/patty/`.

2. SYSTEM OVERVIEW & DESIGN

PATTY takes a text corpus as input and produces a taxonomy of textual patterns as output. PATTY works in four stages:

- **Pattern extraction.** A pattern is a surface string that occurs between a pair of entities in a sentence, thus the first step is to obtain basic textual patterns from the input corpus. We first apply the Stanford Parser [7] to every sentence of the corpus to obtain dependency paths from which textual patterns are extracted.

- **SOL pattern transformation.** The second step is to transform plain patterns into syntactic-ontological-lexical patterns (SOL) patterns thereby enhancing them with ontological types. A SOL pattern is an abstraction of a textual pattern that connects two entities of interest. It is a sequence of words, POS-tags, wildcards, and ontological types. A POS-tag stands for a word of the part-of-speech class such as a *noun, verb, possessive pronoun, etc*. An ontological type is a semantic class name (such as ⟨*singer*⟩) that stands for an instance of that class. An example of a SOL pattern is: ⟨⟨*person*⟩*'s [adj] voice in* * ⟨*song*⟩⟩.

- **Pattern generalization.** The third step is to generalize the patterns, both syntactically and semantically. In terms of lexico-syntactic generalization, patterns are generalized into a syntactically more general pattern in several ways: by replacing words by POS-tags, by introducing wildcards, or by generalizing the types in the pattern. For semantic generalization, we compute synonyms and subsumptions, based on the set of entity pairs the patterns occur with — *support sets*.

- **Subsumption and synonym mining.** The last step is to arrange the patterns into groups of synonyms and in a hierarchy based on hypernymy/hyponymy relations between patterns. For semantic generalization, the main difficulty in generating semantic subsumptions is that the support sets may contain spurious pairs or be incomplete, thus destroying crisp set inclusions. To overcome this problem, we designed a notion of a *soft set inclusion*, in which one set S can be a subset of another set B to a certain degree. We thus produce a weighted graph of subsumption relations between the patterns. Patterns with perfectly overlapping support sets are grouped into synonym sets (synsets), where each such synset represents a single relation.

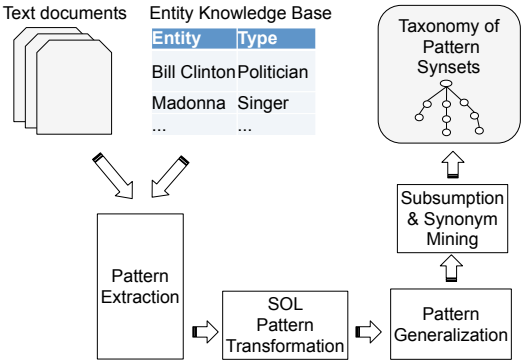


Figure 1: PATTY Architecture

To find entities in the text, and to type them semantically, PATTY requires a pre-defined knowledge base as input. We use either YAGO [13] or Freebase [2]: YAGO has classes derived from Wikipedia categories and integrated with WordNet classes to form a hierarchy of types; Freebase has a handcrafted type system with upper level topical domains as top tier and about entity classes as a second tier. Figure 1 shows the entire PATTY architecture with the role of the knowledge base.

3. IMPLEMENTATION

PATTY is implemented in Java and makes use of the Stanford NLP tool suite for linguistic processing, Hadoop as the platform for large-scale text and data analysis through MapReduce, and MongoDB for storing all resulting data in a key-value representation. The Web-based frontend is running AJAX for asynchronous communication with the server.

Pattern Extraction. The output of pattern extraction are patterns extracted from paths of grammatical dependency graphs, along with the patterns we also output part-of-speech tags of the words from the original sentences. This information is used later for transforming basic patterns into SOL patterns. For distributing pattern extraction with MapReduce, each document is processed independently by the mappers. No coordination is required between concurrent mappers. Thus the input to the mappers are documents from the input corpus. The mapper scans the document, one sentence at a time. If the mapper encounters a sentence with a pair of interesting entities, it emits triples of the form (e_1, p, e_2) along with the necessary part-of-speech information. The MapReduce algorithm is outlined in Figure 3.

SOL Pattern Transformation. We take as input the

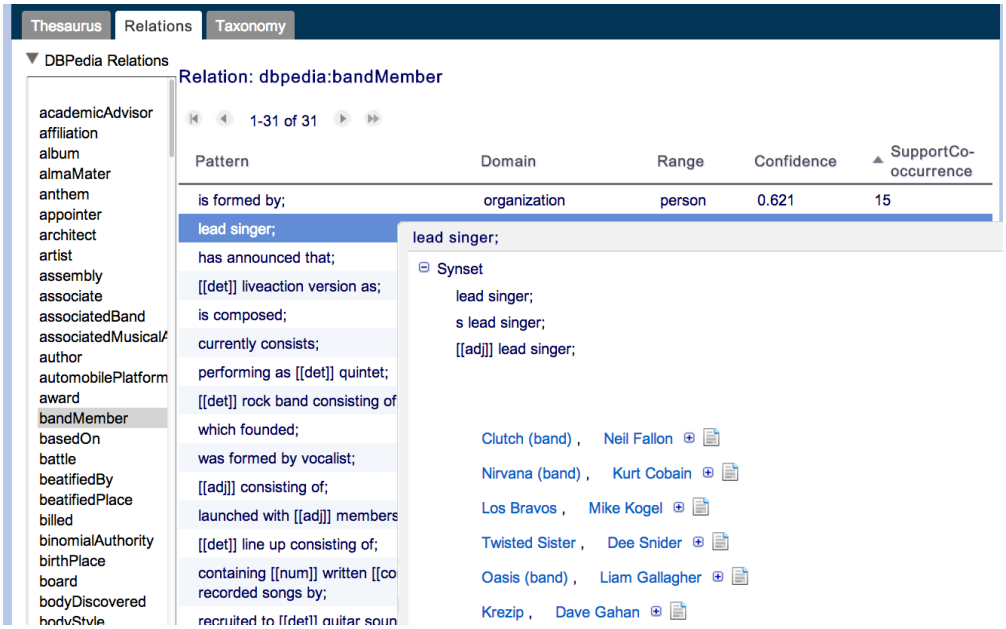


Figure 2: PATTY paraphrases for the DBPedia relation *bandMember*, the type signature and entities occurring with the relation are also displayed.

---

function map(i, d_i)
  List $S \leftarrow$ all sentences from document (d_i)
  for $s \in S$ **do**
    $NE \leftarrow$ detect named entities in s
    if $|NE| > 1$
      $G \leftarrow$ generateDepedencyGraph(s)
      $P \leftarrow$ dependencyPaths($\forall(e_i, e_j) \in NE$)
      for $p \in P$ **do**
        emit(e_i, p, e_j, pos)

Figure 3: MapReduce pattern extraction

basic patterns emitted by the pattern extraction module and emit SOL patterns in the form of a sequences of *n-gram* with *type signatures*. To generate SOL patterns from the textual patterns, we decompose the textual patterns into n-grams (n consecutive words) and then generate type signatures for these n-gram patterns.

Frequent N-gram Mining. Only the n-grams that are frequent in the corpus are retained in the SOL patterns, the rest are replaced by wild-cards. The MapReduce algorithm is outlined in Figure 4. Mappers take basic patterns and generate n-grams and emit, for each n-gram, an intermediate key-value pair consisting of the n-gram and a support of 1. The reducers gather support counts for any given n-gram and sum them up to obtain the final support counts. Only those n-grams whose support is above the specified values are emitted. Once we have the frequent n-grams, a second MapReduce algorithm is used to rewrite patterns into a form with frequent n-grams only, disregarding infrequent ones. This way we end up with n-gram patterns. Next, we generate type signatures for the n-gram patterns.

---

function map(i, p_i)
  List $N \leftarrow$ generateNgrams(p_i)
  for $n_i \in N$ **do**
    emit(n_i,1)

function reduce(n_i, $[v1, v2, v3, ...]$)
  $support \leftarrow 0$
  for $v_i \in [v1, v2, v3, ...]$ **do**
    $support \leftarrow support + v_i$
  IF $support \geq \gamma$ // where γ is minimum support
    emit(n_i, $support$)

Figure 4: MapReduce frequent n-gram mining

Type Signature Generation. For a pattern which is not typed, we can easily compute the occurrence frequencies for each type pair that the pattern occurs with. Based on these initial statistics, we can mine the prevalent type signatures needed to transform type-agnostic patterns into one or more typed patterns.

Given a pattern with type statistics and the entity pairs $(e1, e2)$ in its support set, the key to inferring good type signatures is in the types of entities in a pattern's support

set. We take all types that the knowledge base provides for a given entity and use heuristics to eliminate unlikely type signatures. For every $(e1, e2)$, we create two sets, one for all the types of $e1$, T_{e1} and one for all the types of $e2$, T_{e2}. We then compute the cross-product of the two type sets $T(e1)$ and $T(e2)$ with an occurrence frequency of 1. As we iterate over the entity pairs in the support set, we accumulate the occurrence frequencies for every type signature.

This procedure results in a list of possible type signatures for each pattern. The set of candidate signatures is often very large, so we enforce a threshold on the occurrence frequency and drop all signatures below the threshold.

Subsumption & Synonym Mining. Mining subsumptions and synonyms from pattern support sets is not trivial, because a quadratic comparison of each and every pattern support set to every other pattern's support set would be prohibitively slow. Therefore, we developed a Map-Reduce algorithm for this purpose. As input, our algorithm requires a set of patterns and their support sets. As output, we compute a DAG of pattern subsumptions. We first invert the support sets data. Instead of providing, for a pattern, all entity-pairs that occur with it, we provide for an entity pair all the patterns that it occurs with. This can be achieved by a Map-Reduce algorithm that is similar to a standard text indexing Map-Reduce algorithm.

From this data, we have to compute co-occurrence counts of patterns, i.e., the number of entity-pairs that the supports of two patterns have in common. Our Map-Reduce algorithm for this purpose is as follows: The mappers emit pairs of patterns that co-occur for every entity-pair they occur with. The reducers aggregate co-occurrence information to effectively output the sizes of the set intersection of the possible subsumptions. A single machine version of this algorithm is described in [10, 12].

4. RESULTS

We applied PATTY to different corpora to generate relation taxonomies of varying sizes and quality. The version derived from Wikipedia (ca. 3.8 Million articles, version of June 21, 2011) is the richest and cleanest one. It consists of about 350,000 typed-pattern synsets organized in a hierarchy with 8,162 subsumptions.

Precision. Random sampling-based assessment showed that about 85% of the patterns are correct in the sense that they denote meaningful relations with a proper type signature. Furthermore, the subsumptions have a sampling-based accuracy of 83% and 75% for top-ranked and randomly sampled subsumptions respectively. To further evaluate the usefulness of PATTY, we performed a study on relation paraphrasing: given a relation from a knowledge base, identify patterns that can be used to express that relation. We found paraphrasing accuracy to vary from relation to relation: in some cases as low as 53%, and in others as high as 96%, the results are shown in Table 1 with 0.9-confidence Wilson score interval. A random sample of 1000 paraphrases showed an average precision of 0.76 ± 0.03 across all relations.

Recall. Without a reference resource in the form of a comprehensive collection of relations, their synonyms and subsumptions, evaluating recall is not truly possible. We estimated recall by manually compiling an approximate reference resource in the music domain. The reference resource contains all binary relations between entities that appear in Wikipedia articles about musicians. Out of 169 ground-truth relations, PATTY contains 126.

Scalability. In terms of run-times, the most expensive part is pattern extraction, where we identify pattern candidates through dependency parsing and perform entity recognition on the entire corpus. This phase runs about a day for Wikipedia on a Hadoop cluster with ten Dell PowerEdge R720 machines and a10 GBit Ethernet connection. Each machine has 64GB of main memory, eight 2TB SAS 7200 RPM hard disks, and two Intel Xeon E5-2640 6-core CPUs. On the same cluster, all other phases take less than an hour to execute.

5. APPLICATIONS

The data produced by PATTY is a valuable resource for a variety of applications. First, it can boost IE and knowledge base population tasks by its rich and clean repository of paraphrases for the relations. Second, it can improve Open IE by associating type signatures with patterns. Third, it can help to discover "Web witnesses" when assessing the truthfulness of search results or statements in social media [5]. Last, it provides paraphrases for detecting relationships in keyword queries, thus lifting keyword search to the entity-relationship level. This can help to understand questions and text snippets in natural-language QA.

We developed a front-end to the PATTY data for exploring these possibilities in three ways: (1) using PATTY as a thesaurus to find paraphrases for relations, (2) using PATTY as a simple kind of QA system to query the database without having to know the schema, and (3) exploring the relationships between entities, as expressed in the textual sources. The Web-based front-end is running AJAX for asynchronous communication with the server.

5.1 Using PATTY as a Thesaurus

PATTY connects the world of textual surface patterns with the world of predefined RDF relationships. Users who are aware of RDF-based knowledge bases can explore how RDF relations map to their textual representa-

Relation	Paraphrases	Precision
DBPedia/artist [*musical_composition* × *musician*]	83	0.96±0.03
DBPedia/associatedBand [*musician* × *organization*]	386	0.74±0.11
DBPedia/doctoralAdvisor [*person* × *person*]	36	0.558±0.15
DBPedia/recordLabel [*musician* × *organization*]	113	0.86±0.09
DBPedia/riverMouth [*river* × *location*]	31	0.83±0.12
DBPedia/team [*athlete* × *team*]	1,108	0.91±0.07
YAGO/actedIn [*actor* × *movie*]	330	0.88±0.08
YAGO/created [*entity* × *entity*]	466	0.79±0.10
YAGO/isLeaderOf [*person* × *organization*]	40	0.53±0.14
YAGO/holdsPoliticalPosition [*person* × *person*]	72	0.73±0.10

Table 1: Relation Paraphrasing Precision for Sample DBPedia and YAGO Relations

tions. For example, as shown in Figure 2, PATTY knows about 30 ways in which the DBPedia relation *bandMember* can be expressed textually. We hope that this wealth of data can inspire new applications in information extraction, QA, and text understanding.

Users do not need to be familiar with RDF in order to use PATTY. For example, users can find different ways to express the *hasAcademicAdvisor* relation, simply by typing "worked under" into the search box. PATTY also provides the text snippets where the mention was found as a proof of provenance. These text snippets can be explored to understand the context in which a pattern can have a certain meaning. In addition, users can browse the different meanings of patterns, as they occur with different types of entities.

5.2 Schema-Agnostic Search

Internally, PATTY stores all extracted patterns with their support sets. This allows users to search for facts in the database. For this purpose, the PATTY front-end provides a search interface where the user can enter Subject-Predicate-Object triples. Different from existing systems, the user does not have to know the schema of the database (i.e., the relations of the fact triples). It is fully sufficient to enter natural language keywords. For example, to find the co-stars of Brad Pitt, the user can type "costarred with" in place of the relation. PATTY will then search not only for the exact words "costarred with" but also automatically use the paraphrases "appeared with", "cast opposite", and "starred alongside". This way the query needs to be issued only once and the user does not need to enter multiple paraphrases. For each result, PATTY can show the textual sources from which it was derived.

The type signatures of the patterns can be used to narrow down the search results according to different semantic types. For example, when searching for a popular subject like Barack Obama or Albert Einstein, the result may span multiple pages. If the user is interested in only one particular aspect of the entity, then the domain of the subject can be semantically restricted. For example, to see what PATTY knows about Albert Einstein in his role as a scientist, the user can restrict the domain of the relation to *scientist*. Such a query returns Einstein's teaching positions, his co-authors, information about his theories, etc.; but it does not return information about his wives or political activities.

These schema-agnostic queries can be extended to simple join queries. This works by filling out multiple triples and linking them with variables, similar to the way SPARQL operates. Different from SPARQL, our system does not require the user to know the relation name or the entity names. For example, to find visionaries affiliated with MIT, it is sufficient to type: *?x vision ?y, ?x ?z MIT*. This will search for people $?x$ who have a vision $?y$ and who stand in some relationship $?z$ with an entity with name *MIT*. These returns figures like Vannevar Bush (The Endless Frontier vision) and Tim Berners-Lee (Web vision).

5.3 Explaining Relatedness

PATTY can also be used to discover relationships between entities [5]. For example, if the user wishes to know how Tom Cruise and Nicole Kidman are related, it is sufficient to type "Nicole Kidman" into the subject box and "Tom Cruise" into the object box. PATTY will then retrieve all semantic relationships between the two, together with the patterns in which this relationship is expressed. For each result, users can click on the source button discover provenance.

This principle can be extended to full conjunctive queries. For example, to find the entity that links Natalie Portman and Mila Kunis, the user can type: *Natalie Portman ?r ?x, Mila Kunis ?s ?x*. This will find all entities $?x$ that link the two actresses, as well as an explanation of how this entity establishes the link. In the example, PATTY finds the movie "Black Swan" for $?x$, and says that both actresses appeared in this movie. As this example shows, PATTY has created an internal, semantic representation of the input text documents, which allows it to answer semi-structured queries. In addition,

to generate semantic patterns, PATTY has implicitly summarized the input text documents. Users can exploit and query these summaries.

5.4 Other Use Cases

Recently, followup work has shown successful usage of PATTY for other tasks. In [9], PATTY 's type signatures are used for semantic typing of out-of-knowledge-base entities. Because the type signatures are fined-grained (e.g., musician, journalist, etc.), the application infers more semantically informative types than standard named entity recognition which works with coarse types such as company, person, etc. In [16], PATTY's relation paraphrases are used for question understanding in the challenging task of question answering.

6. RELATED WORK

Recently, [8] and [17] have addressed the mining of equivalent patterns, in order to discover new relations, based on clustering. These approaches are based on building large matrices or inference on latent models. They differ from PATTY in that the issue of identifying subsumptions between patterns has been disregarded. Among prior works, only ReVerb[4] and NELL[3], have made their patterns publicly available. However, the ReVerb patterns for Open IE are fairly noisy and connect noun phrases rather than entities. NELL is limited to a few hundred pre-specified relations. None of the prior approaches knows the ontological types of patterns, to reveal, e.g., that *covered* holds between a musician and a song.

7. FUTURE WORK

There are several avenues for future research that can build on and improve PATTY. We focused on two types of relatedness: synonymy and hypernymy. However, further types of relatedness between binary relations can be extracted. For example, we can also extract antonyms, where one relation is the opposite of another. Some relations have units; so we could extract the units of relations such as *hasHeight, hasRevenue, hasLength (for songs),* etc. In addition, some relations have value constraints, for example, it is not possible for a person's height to be 5 meters. Another line of future work is extracting n-ary relations for $n > 2$. Such relations might be better suited for explaining complex events and causality.

8. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z.G. Ives: DBpedia: A Nucleus for a Web of Open Data, ISWC/ASWC, pp. 722-735 2007

[2] K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor: Freebase: a Collaboratively Created Graph Database for Structuring Human Knowledge. SIGMOD, pp. 1247-1250, 2008

[3] A. Carlson, J. Betteridge, R.C. Wang, E.R. Hruschka, T.M. Mitchell: Coupled Semi-supervised Learning for Information Extraction, WSDM, pp. 101-110, 2010

[4] A. Fader, S. Soderland, O. Etzioni: Identifying Relations for Open Information Extraction, EMNLP, pp. 1535 - 1545, 2011

[5] L. Fang, A. Das Sarma, C. Yu, P. Bohannon: REX: Explaining Relationships between Entity Pairs. PVLDB 5(3), pp. 241-252, 2011

[6] G. Limaye, S. Sarawagi, S. Chakrabarti: Annotating and Searching Web Tables Using Entities, Types and Relationships. PVLDB 3(1), pp. 1338-1347, 2010

[7] M.-C. de Marneffe, B. MacCartney and C. D. Manning. Generating Typed Dependency Parses from Phrase Structure Parses. LREC, 2006

[8] T. Mohamed, E.R. Hruschka, T.M. Mitchell: Discovering Relations between Noun Categories, EMNLP, pp. 1447-1455, 2011

[9] N. Nakashole, T. Tylenda, G. Weikum: Fine-grained Semantic Typing of Emerging Entities, ACL, *to appear* 2013.

[10] N. Nakashole, G. Weikum, F. Suchanek: PATTY: A Taxonomy of Relational Patterns with Semantic Types, EMNLP, pp.1135 -1145. 2012

[11] N. Nakashole, G. Weikum, F. Suchanek: Discovering and Exploring Relations on the Web. PVLDB 5(10), pp. 1982–1985, 2012

[12] N. Nakashole: Automatic Extraction of Facts, Relations, and Entities for Web-Scale Knowledge Base Population. *PhD Thesis, Saarland University*, 2012

[13] F.M. Suchanek, G. Kasneci, G. Weikum: Yago: a Core of Semantic Knowledge, WWW, pp. 697-706, 2007

[14] P. Venetis, A. Halevy, J. Madhavan, M. Pasca, W. Shen, F. Wu, G. Miao, C. Wu: Recovering Semantics of Tables on the Web, VLDB, pp. 528-538, 2011

[15] W. Wu, H. Li, H. Wang, K. Zhu: Probase: A Probabilistic Taxonomy for Text Understanding, SIGMOD, pp. 481- 492, 2012

[16] M. Yahya, K.Berberich, S. Elbassuoni, M. Ramanath, V. Tresp, W. Weikum: Natural Language Questions for the Web of Data. EMNLP, pp. 379-390, 2012

[17] L. Yao, A. Haghighi, S. Riedel, A. McCallum: Structured Relation Discovery using Generative Models. EMNLP, pp. 1456 -1466, 2011

Jeff Vitter Speaks Out on being a Southerner, Duties of a Dean, and More

by Marianne Winslett and Vanessa Braganholo

Jeffrey S. Vitter
http://provost.ku.edu/jsv

Welcome to this installment of ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today I am at Purdue University. I have here with me Jeff Vitter, who is the Frederick L. Hovde Dean of the College of Science[1]. Before coming to Purdue, Jeff was on the faculty of Duke and Brown for many years, and he served as the chairman of the Department of Computer Science at Duke. Jeff's research interest lies in algorithms, especially in the areas of external memory algorithms and compression. Jeff is an ACM Fellow, IEEE Fellow, and Guggenheim Foundation Fellow[2]. He is on the board of directors of the Computing Research Association and is the former chair of ACM SIGACT. His PhD is from Stanford. So, Jeff, welcome!

Great! Thanks for having me, Marianne.

[1] This interview was conducted in 2008. Today, Jeff Vitter is the provost and executive vice chancellor and the Roy A. Roberts Distinguished Professor at the University of Kansas.

[2] In 2009, Jeff was elected as a Fellow of the American Association for the Advancement of Science (AAAS).

Jeff, what was it like working with Don Knuth at Stanford?

Don is just an incredible human being. You know, he is really probably more responsible than any other person for the founding of computer science as an academic discipline. So just getting his insights was really tremendous. Professionally, what really impacted me was his sense of importance of theory and practice, and how it's vital to have a deep understanding of them both in order to excel at either. It was a little intimidating because he had just started TeX, and he really wasn't taking students, and one day I went to see him and I told him I had solved this problem and I thought this other one might be interesting to look at, just to see what he thought. And he said, "Well, if you do that, that would make a great thesis. And, by the way, you should plan to do this here, then this then, and graduate at this time," which was in three years from when I got to Stanford. I didn't dare question this. I just plowed ahead and did it. And I remember going through my thesis near the end of my third year, getting ready to finish, and Don looks at me and says, "You know, you really did quite a bit here, in an amazingly short amount of time. Why did you do it so quickly?" And I am sitting there after having worked so hard, and I was about to say, " 'Cause you told me to!" (Laughing.) But it was just a great experience. He was the most remarkable academic I have ever met.

So, what was that thesis on?

It was on Coalesced Hashing, as it's called. It is a hashing method that optimizes the way it uses storage in order to get the absolute best in search time. I have adopted the name "Coalesced" for some our projects here in the College of Science.

Jeff, most of your research is on algorithms for massive data sets. But your papers mainly appear in theory-oriented venues like Algorithmica and FOCS (Foundations of Computer Science), rather than SIGMOD, VLDB, and ICDE. So are you a theory guy, a database guy, or a database theory guy?

Yes (grinning).

To follow up on what I learned from Don, I think the most important things are this blending of theory and practice, so that is what I try to instill in my students. I really try to cover both of those communities. I have had some great students who have gone on in the systems arena, but because they have such a strong theory background and can appreciate the elegance and essence of what the techniques they are working on are all about, I think that really brings a scalability that makes what they do in systems work out. I have had students like Mark Nodine, Paul Howard, Dzung Hoang, Darren Vengroff, Lipyeow Lim, Tavi Procopiuc, Rakesh Barve, and Min Wang. They are incredible systems implementers, but they are also fundamentally very strong algorithmic students. I think that is part of the reason they are so good in systems.

What is the relationship between compression and database query optimization?

Historically, histograms are used a lot in order to summarize what's happened in the past to guide decisions for query execution or whatever. And my interests in this field, I really have a variety of different interests, and that is really what drives me as a researcher. One of my grad students, Min Wang, and I were working in the area of looking at compression because I was looking at compression from a variety of fronts, and along with Yossi Matias, we collaborated on applying wavelets. It was really the first time wavelets were used in the database community. It was used in a way to really be a novel form of histogram; capturing data in a fundamentally more efficient way, more effective way. So we worked out a lot of algorithmic aspects, it was very effective for doing this kind of query estimation we are talking about, or doing approximate answers, if you are in OLAP-type query situations. That has led to a lot of other work where wavelets have proven to be very effective. There have been great results by some on how to get provable bound estimates through wavelets. So, that has been a very exciting thing. But you know, the goal of all of these areas is really prediction. If you can do a better job of predicting what will happen in the future, you are going to be able to have a more effective system, more efficient or whatever.

Academic administration […] is really computer science on a grander scale. It is problem solving, or to put it more positively, it is finding solutions

Prediction is really nothing more than learning. It is trying to understand what will happen. That has driven a lot of my fundamental research. So, to give you an example, let's take a learning problem, which is the same as prediction, trying to learn what an elephant is. Suppose I want to teach you what an elephant looks like. This is actually very relevant in this U.S. Presidential election year, cause a lot of people are trying to understand what elephants look like. So here is the problem, I am going to give you a bunch of photos of animals, and I am going tell you for each one if it is an elephant or if it is not an elephant. And after a while, hopefully if you are a good learner, you will be able to know what an elephant is. So if I give you a new picture, that you haven't seen before, you will be able to tell if it is an elephant or not correctly. So, in the computational learning theory area, there is a domain called "PAC learning,"[3] where you can actually prove that learning is the same thing as data compression in the intuitive sense that if you as a learner do nothing more than memorize the pictures I showed you, you are going to have no chance of then classifying this new picture. But if instead, you have compressed what you have seen into a few basic rules, like elephants are grey, they are big, they have a trunk, they do not have wings, and things like that, then you will have no trouble classifying the

[3] PAC learning stands for probably approximately correct learning.

new picture as to whether it is an elephant. And that is really the essence of this relationship.

So, we were looking at a variety of problems, one of them was prefetching. Prefetching is a job where you have a bunch of accesses to a disk in the past, and now based on those you want to predict what are you going to access in the future so you can prefetch it into memory and have it ready for when you are going to access it, and avoid a costly page fault. So we applied a data compression method because of this intuition that compression is really prediction. We applied a data compression method to the sequence of numbers, which are page accesses, and in the bowels of the method, the Lempel-Ziv method, was a prediction for what the next page reference would likely be. We used that, and we showed that actually it allowed us to boost the hit rate from 20% in many applications, up to 70%, so it was very effective. And it has a really nice mathematical foundation. So prediction and compression come into play in a lot of instances. In image databases, it's the key for storing images so that you can search for them based on similarity. And of course, any time you have compressed data, it will often be stored in faster areas in the memory hierarchy, and then it makes it more efficient.

You wrote the book — literally[4] — on external memory algorithms. What are they, and how do they relate to databases?

It all goes back to a model of memory hierarchies, or what we call a parallel disk model, where in a simple setting, we have a computer with an internal memory and data are simply too large to fit in the internal memory, so we store it on disk. And this is a standard database set-up. Because disk drives are these physical rotating media where it takes milliseconds to get to data, but once you get to data, you can get adjacent data very quickly, the result is that data are typically transferred in blocks because that amortizes the cost of the high latency just to go to the data. One of the main goals of external memory algorithms is to minimize the number I/O transfers. And I/O is transferred in large blocks of data, so the main parameters of the model are the size of the transferred block, the size of the internal memory, and then basically that's it, the problem size itself. And the goal is to design an algorithm that uses locality in a fundamental way, so that data are transferred in blocks, and when you want data, you want a block of data, you don't want data from random locations, because if you do things effectively, you can speed up computations by a factor of 100 or 1,000 because of this block mechanism. So to give you an example, we applied this in a domain at Duke in collaboration with some folks in the School of the Environment. Lars Arge and I and students and collaborators in the School of the Environment worked on methods for determining, when rain falls, where it will go. So, what will the watershed be? Where will the flooding occur? This is very

[4] J. S. Vitter. *Algorithms and Data Structures for External Memory*, Series on Foundations and Trends in Theoretical Computer Science, Now Publishers, Hanover, MA, 2008. Also published as Volume 2, Issue 4 of *Foundations and Trends in Theoretical Computer Science*.

important in North Carolina. So we took satellite data and other imaging methods of regions like the Appalachians, and using so-called conventional techniques, such as ArcInfo, these calculations could take several days. There would be calculations that could not be run at all. Using newly-designed algorithms that focus on block transfer, we were able to reduce the running time from days to hours, or when they couldn't even be computed at all, we could do them in just a few hours. So it can make a really big difference, especially because data are just expanding at a crazy rate.

You are a relatively recent transplant from the east coast to the Midwest. What do you think of life in the Midwest?

I grew up in the south, went to grad school in California, and then I was at Brown and Duke on the east coast. But I did go to Notre Dame as an undergrad, so I have strong roots in Indiana. I am happy to say that being two hours south makes a big difference in temperature. It is a lot warmer and more moderate here. The main thing about Indiana is it is a great family environment. West Lafayette in the last 10 years has gotten some really wonderful restaurants, culture opportunities; in fact, there is a New Orleans restaurant that just opened a couple of months ago, and the owner and chef is a high school classmate of my brother Mark, so it is really good. It's a great place to live. And the students here are, with their Midwestern ethic, just very hard workers. They are wonderful to work with.

[…] in the arena of the life sciences and biology, there are great opportunities that put databases at the forward.

Some people think that CS researchers who aren't on the east or west coasts must be quite isolated. Have you found that to be true?

It is a perception that is challenging at recruiting time, but when you show the candidates all that is going on, all that we have at Purdue, it is really quite remarkable. In databases, with this community, we have an incredible group. We have Ahmed Elmagarmid, Walid Aref, Elisa Bertino, Chris Clifton. It's a great group. Ahmed is actually the head of the Cyber Center, which integrates IT research across the entire University. In information security, we have what I think is the best group anywhere. 25% of all of the information security PhDs in the entire country come out of Purdue and our CERIAS Center. Mike Atallah and Gene Spafford are just renowned in that area. We have terrific systems people, whether it is in networking, distributed systems, or programming languages, operating systems, graphics and visualization, software engineering. It is really a strong group. So this is a great place to be, and I am very excited to be here.

What about your interactions with other Universities?

That is a great thing, because the CIC or the Big Ten has universities that very closely collaborate. In fact, Marianne, you just drove over in an hour and a half from Illinois. We have great collaborations with Illinois, Michigan, of course. We are two hours from Chicago, so it is an opportunity to work with many researchers. I mentioned the ones at Purdue, but the whole region is quite a rich area, and a great place for people to thrive in databases.

What led you to get an MBA in 2002?

When I went to Duke, which was to become department chair, it was just a great experience. It was an experience of building a new department culture, fundamentally based on getting everybody involved from the students on up and energizing it to really move from where it was to the great department it is today. In the process, I got very interested in academic administration, which I think is really computer science on a grander scale. It is problem solving, or to put it more positively, it is finding solutions. And I wanted to get a more formal background. An MBA was really an eye opening experience, because it is a new culture, you are learning new tools, and it was just fascinating to me, especially this notion of strategic planning, which is so important for what we are doing now. So, I just had a great time there. Plus, the Fuqua School at Duke has absolutely the best food in Durham, and we could eat all we wanted, so it was worth it just for that alone.

You mean the MBA students have free food?

Yep, they sure do.

Maybe we should try that in Computer Science.

Well, it might be costly, if you have ever seen the grad student receptions, but I am sure it would be effective.

So, how did you have time to do the MBA while you were also chair of the department?

I timed it so that it was near the point that I was going to step down, so I really overlapped just a semester that way. Then, fortunately, I taught half-time during the following year, so it really worked out well. It was a lot of work, but it was a great experience.

Has your MBA been useful?

Oh, definitely. One thing is just the way that it helps you look at problems and situations and understand the inner relationships, but just thinking strategically and long-term and how you need to really focus on what is going to count down the road because when you get there you cannot go back and change things years ago. We are in the midst of strategic planning now, and one of the things we did that was really

fundamental that I think is quite unique across the country is that we have instituted a way of dealing with these large multi-disciplinary problems that are society-wide: trying to find new forms of energy, trying to deal with the climate change and the environment situation, trying to cure and prevent disease. These are problems that require contributions from multiple disciples; certainly computer scientists, but from all over. They just were not getting proper attention, because we were doing things discipline by discipline, and we were focusing on hiring faculty who were going to be the best for our individual disciplines. And in fact, if a faculty wanted to work elsewhere and collaborate, they were almost seen as perhaps a department losing half of a slot, so we wanted to allow departments who had these priorities already to be able to realize them.

We spent a year determining the priorities, but we also had a mechanism in place so that as we were growing — and Purdue was growing by 300 faculty, 60 in our College of Science — and filling these positions, we adopted the approach that we were going to devote these multidisciplinary priorities as the key for these growth positions. We did college-wide searches for these areas, and it's become so much a culture now at our college that as we near our steady state in faculty size, we have decided this is something we want to continue, but we have to do it by a different mechanism. The MBA experiences now help me help design the new mechanism because it is a different circumstance; you cannot use the old approach. You have to design something that makes sense for the time. So we have that, it is unique, it's for our current situation, but it is allowing us to continue this multidisciplinary momentum. So that is what an MBA can help do.

You're now in your sixth year as Dean of Science here at Purdue. What do deans do?

Well, our fundamental mission is to help faculty, students, and staff succeed, so that is my number one goal; and it is through visioning and strategic planning like I talked about. It is raising money. It's trying to be careful in budget management so we can spend money for the things that are important. It's designing curriculum. It is really helping people succeed, fundamentally.

But everything you have just said, at least at Illinois, is also the job of a department head.

That is true, but deans have a broader responsibility. They need to help facilitate the interactions between departments, which is really a substantial challenge. It takes a lot of collaboration and listening. You have really got to communicate and talk a lot with people to understand where they are coming from, what they want to do, and how you can best help them succeed. It is a big job, but it is really fascinating, because when things work, they can have a dramatic effect on people, on lives, on jobs, on revitalizing a state's economy, hopefully leading this country to a brighter future.

You have 5 papers in DBLP for 2007, and more than that for the previous year. How can you be a dean and still be doing research?

So what you are saying is that I am actually publishing less as the years go on, is that what you are saying, Marianne? (Laughing.)

It actually goes up and down, so I don't think we can just extrapolate linearly.

I think the most important thing is to go and talk with your colleagues in physics, chemistry, biology, history, music, other parts of engineering, because they are just ripe for applications and new kinds of insights that will help motivate new things.

I think, to me, I love research. But more fundamentally, I think it makes me more in tune with what is going on in the college. Staying involved in research keeps me vital. Faculty work incredibly hard, they have a lot of things pulling them in different directions, and I think I should at least work as hard as they do, because we have such a great group here.

People always point to the physicists saying how effective they are at working together to get funding for their research. Computer Scientists tend not to do things too often as a body, or speak with one voice.

In fact, they often shoot each other! I guess that is a way of having one voice: if you shoot each other, there is only one person left. Astrophysicists, for example, are renowned at getting together, deciding what are the key often instrumentation needs that they have that will enable the great things they want to do. Then, in a single voice, they lobby and get those sorts of things. That is really what the CCC is all about. Ed Lazowska is leading that effort in the CRA. It is very important to our future because we need absolutely to get that message out. We need to address the pipeline issue. We are seeing slightly higher enrollments now, but we are 50% under nationally in enrollments in computing than we were just six years ago. It is quite a problem. So we have to get the pipeline in because when you look at the *Gathering Storm* report that came out of the National Academy, there is a tremendous need, and computing has one of the most opportunities for jobs of any discipline. We have 150,000 new jobs created each year, and we graduate 50,000 students.

You wouldn't know it to read the newspapers, would you? They always talk about off-shore jobs.

Exactly, I think it's parents telling their kids, "Don't major in computing because the jobs are going overseas." So we are trying to get the message out that it is actually

the opposite. And unless we do something, we are going to be struggling in this country, and the biggest place we can make a mark is in the under-represented groups. For women, we are down tremendously for women going into computing these days, and minorities, such as African-American, Hispanics, and Native Americans, we need to do a much better job. And southerners too.

Many people think that computer science as an academic discipline will wither away like railroad engineering: today, you don't see Departments of Railroads in universities. Recently, computer science has been moving closer to its application domains, and you can see this trend especially clearly in the database world. Are we going to wither away and be absorbed by these application areas?

I hope not. And I think the key to being a vital field is to actually embrace those connections and make them a fundamental part of what we do. The real value of multidisciplinary opportunities is, first of all, that they solve the big problems, not artificial problems. Secondly, the most effective outcome is when you really make deep contributions within each discipline as part of this collaboration. And in the course of working on these problems, you will have suggested to you fundamental problems in your discipline, and that is what keeps disciplines alive. If computer science can really embrace this collaborative role it has with other disciplines, it will be revitalized by the very issues that those other disciplines suggest, and that will always keep computer science as a very strong force that will warrant and have people's appreciation.

The way you say that, it almost sounds like the other fields will inspire us by suggesting what direction we should be going, rather than CS having the intellectual leadership.

Well, it is a collaboration, and I think it takes the trust and willingness to not be concerned about who suggested what, so that you can just drive forward, and collaboratively both groups — application arena groups and CS people — are going to make fundamental contributions. If we don't do that, I think what will happen is that the other disciplines will recognize the need for it on their own and adopt computing in their disciplines, and I think that's what the real danger is to computing. So we have an opportunity to revitalize computing by embracing all of these opportunities.

What are the most challenging database issues in other scientific disciplines?

I think in the arena of the life sciences and biology, there are great opportunities that put databases at the fore. For example, in biology, I just have to mention that here at Purdue we have what I think is the top structural biology group in the world. They are focused on understanding the geometry of macromolecules, whether they are viruses or nucleic acids, or whatever, because, in biology, form often determines function. If you take this virus, and you can understand its structure, then drug designers can design drugs that bind to it just right to block its function and cure the disease. Bringing geometry in a fundamental way into databases is really an

important challenge — and a very necessary one for this huge area of life sciences. I think that is a great opportunity. Other applications where, for example, satellite data come so fast, suggest new ways of approaching databases, like data streaming, those are interesting aspects too. So I think there are a variety of ways where databases can grow into new areas.

Very few computer science researchers come from the deep south in the US — although you and I are two exceptions. What does your southern background mean to you?

Well, as you know southerners have just in them an identity, and it is especially true in New Orleans because of the very distinct culture that is quite different from the rest of Louisiana, for example. So I will always consider myself a southerner. I am concerned. I think the south has suffered because it is not participating in the high-tech revolution that other parts of the country are really deeply involved in. We need to reverse that. We need to get all under-represented groups involved because we have this great shortage, and this is an opportunity to try to tap into the south and get them focused. So, as a southerner, I feel a lot of regional pride, but also concern, and I hope we can help reverse that situation.

Our fundamental mission [as deans] is to help faculty, students, and staff succeed.

So when you talk about tapping into it, do you mean we should take those southerners and bring them up north and educate them in the ways of computers, or are you talking about a revolution from within?

Certainly at southern universities there are great opportunities to develop a more substantial database presence, and in general computer science. That, it think, will be very important. As they develop new technologies, they are going to need that environment. Richard Florida is an author who has this thesis that the great economic centers are fundamentally built around great universities because creative people are attracted to places that are vital in culture. We have to build that in the south, and I think it will all come together.

Southerners are attracted to places with great football, so maybe that is key.

That's true. I went to Notre Dame as an undergrad, which is an archenemy of most schools in the south, but it was a fun rivalry.

So, if you have this strong southern identity, where is your strong southern accent?

Well, I have no doubt lost some of it. The best book to get an understanding of real New Orleans is *A Confederacy of Dunces*, by John Kennedy Toole. In the forward of this book, there is a little blurb from probably a hundred years ago that describes a

New Orleans accent as really a soft Brooklyn accent. And that is really what it is. If you go to New Orleans, if you hear a southern accent, it is certainly someone who wasn't born there. But a real New Orleans accent is a real Brooklyn type accent.

Can we get a demonstration here? I'm not quite following you.

Well, if I saw you at the local drug store (and of course you'd have your hair up in curlers), I'd say (in New Orleans accent), "Hey, where y'at, MariANNE? Whatcha doin'? You wanna go get some red beans and rice?"

There is that tang in there.

But it is nothing like a southern accent. In fact, an expression in New Orleans for "how are you?" is "where y'at?" New Orleanians are called Yats as a result. That's the name of the restaurant that just opened here in West Lafayette; the New Orleans restaurant is called Yats.

Do you have any words of advice for fledgling or midcareer database researchers or practitioners?

I think the most important thing is to go and talk with your colleagues in physics, chemistry, biology, history, music, other parts of engineering, because they are just ripe for applications and new kinds of insights that will help motivate new things.

If you magically had enough extra time to do one additional thing at work that you are not doing now, what would it be?

Actually, it would be to go home and spend more time with my family and kids. I have an incredible wife, Sharon, and three wonderful kids, Jillian, Scott, and Audrey. I just wish I could say I was more responsible than I am for how they have turned out. So I would spend more time at home.

If you could change one thing about yourself as a computer science researcher, what would it be?

I just wish I had the time to learn more things, because there are so many fascinating connections, and many things that I do are dealing with applying paradigms or insights that I picked up one place that shed a new light in another domain and lead to interesting new results. I just wish I had the opportunity to learn more things and keep up with all the things going on in computing and other fields.

Well, thank you very much for talking with me today.

Great, it was a pleasure to be with you. Thank you.

Database Research at the National University of Singapore

Stephane Bressan Chee Yong Chan Wynne Hsu Mong-Li Lee
Tok-Wang Ling Beng Chin Ooi Kian-Lee Tan Anthony K.H. Tung
National University of Singapore, Singapore 117417

1. INTRODUCTION

At the National University of Singapore (NUS), the database group has worked on a wide range of research, ranging from traditional database technology (e.g., database design, query processing and optimization) to more advanced database technology (e.g., cloud and big data management) to novel database utilities (e.g., database usability, visualization, security and privacy). In this article, we describe some recent and on-going interdisciplinary projects for which we have received significant amount of funding.

2. CLOUD-BASED DATA MANAGEMENT

We have been developing efficient cloud computing platforms for large-scale services, and Big Data management and analytics using commodity hardware. We shall elaborate them below.

2.1 MapReduce-based Systems

One of our goals is to allow users of MapReduce-based systems to keep the programming model of the MapReduce framework, and yet to empower them with data management functionalities at an acceptable performance. We achieved this in two directions. First, we sought to identify key design factors of MapReduce (Hadoop) that affect its performance [17]. We conducted a comprehensive and in-depth study of Hadoop, and found that, by carefully tuning these factors, we can achieve much better performance. For example, MapReduce can benefit much from the use of indexes, and its performance can improve by a factor of 2.5 for selection tasks and a factor of up to 10 for join tasks. We also showed that, among the two types of I/O interfaces for scanning data, the direct I/O mode is superior over the streaming I/O mode.

Second, we have developed query processing engine under the MapReduce framework. At the operator level, we have developed join algorithms. In particular, our proposed MapReduce-based similarity (kNN) join exploits Voronoi diagram to minimize the number of objects to be sent to the `reducer` node to minimize computation and communication overheads [25]. We also designed several schemes for processing multi-join queries efficiently - while the Map-Join-Reduce mechanism [18] introduces a `join` operator to combine multiple datasets, the multi-join scheme in AQUA [40] exploits replication to expand the plan space. We have also developed an automatic query analyzer that accepts an SQL query, optimizes it and translates it into a set of MapReduce jobs [40]. Finally, to support data warehousing, we have leveraged on column store, and proposed Concurrent Join to support multi-way join over the partitioned data [20]. In all these works, we target to reduce the number of MapReduce jobs to minimize the initialization overheads.

2.2 epiC: A V^3-aware Data Intensive Cloud System

Our second direction is driven by the limitations of MapReduce-based systems to deal with "varieties" in the cloud data management. Most business production environments contain a mixture of data storage and processing systems; for example, customer data are maintained by a relational database and user requests are logged to a file system, while images and digital maps are handled by an object storage system. Processing and analyzing these data often requires different APIs and tools. SQL may be used for generating reports, while proprietary libraries may be used for feature extraction from images. Therefore, migrating such federated production systems into a centralized cloud infrastructure introduces three kinds of varieties (called V^3): variety of data (e.g., structured and unstructured), variety of storage (e.g., database and file systems), and variety of processing (e.g., SQL and proprietary APIs).

The V^3 problem mentioned above poses two main challenges to the cloud data management system: resource sharing and heterogeneous data process-

ing. It is well known that deploying multiple storage systems on the same cloud can increase the utilization rate of underlying hardware since spaces released by one system can be reclaimed by another. However, the challenge is how to guarantee the performance isolation. For example, systems like HDFS or GFS are optimized for large sequential scanning and thus prefer manipulating large files. Sharing disks between such systems with the key-value stores may degrade their performance since key-value stores frequently create and delete small sized files, resulting in disk fragmentation.

MapReduce system is proven to be highly scalable for large scale data processing. But the system requires its users to re-implement their existing data processing algorithms with MapReduce interfaces. As an example, one must implement an SQL engine on top of MapReduce in order to perform SQL data processing. Such problem is not trivial for federated production systems, where multiple data formats have to be supported.

As a response to the V^3 challenge, we initiated the epiC project, a joint system project between researchers from NUS and Zhejiang University [2]. The goal of epiC is to provide a framework for facilitating companies to deploy and migrate their federated data systems to the cloud. The epiC system adopts an extensible design. The core of epiC provides two services: virtual block service (called VBS) which manages the cloud storage devices and a coordination framework (called E^3 [9]) which coordinates independent computations over federated systems. To analyze the data, users invoke a set of computing units (called Actors). In each Actor, users employ their favorite APIs to process a specific type of data and use E^3 to coordinate these Actors for producing the final results.

We have developed a novel elastic storage system (ES^2) [8] and deployed it on epiC. ES^2 employs vertical partitioning to group columns that are frequently accessed together, and horizontal partitioning to further split these column groups across a cluster of nodes. A number of novel cloud-based indexing structures (e.g., B^+-tree [39, 12], bitmap indexes [24], R-tree index [37]) have been developed.

We have also examined how transactions can be supported. This led to the design of ecStore [35]. ecStore exploits multi-version optimistic concurrency control and provides adaptive read consistency on replicated data.

2.3 Peer-to-Peer-based Cloud Data Management

Another direction that we are pursuing is the integration of cloud computing, database and peer-to-peer (P2P) technologies. Exploiting a P2P architecture on a cluster of nodes offers several advantages over the MapReduce framework: (a) It offers more robust query processing mechanisms as nodes can now communicate with one another; (b) It removes the single point-of-failure in the master/slave architecture of MapReduce; (c) It facilitates elastic design as peers can be readily added and removed in a P2P architecture.

BestPeer++. We have developed BestPeer++ [11, 10], a cloud-enabled evolution of BestPeer [26]. BestPeer++ is enhanced with distributed access control, multiple types of indexes, and pay-as-you-go query processing for delivering elastic data sharing services in the cloud. The software components of BestPeer++ are separated into two parts: *core* and *adapter*. The core contains all the data sharing functionalities and is designed to be platform independent. The adapter contains one *abstract* adapter which defines the elastic infrastructure service interface and a set of *concrete* adapter components which implement such an interface through APIs provided by specific cloud service providers (e.g., Amazon). We adopt this "two-level" design to achieve portability. BestPeer++ instances are organized as a structured P2P overlay network. We have used BATON [16], developed at NUS, as it can support range queries efficiently. The data are indexed by the table name, column name and data range for efficient retrieval.

Katana. The *Katana* framework is a novel peer-to-peer (P2P) based generalized data processing framework [14]. It can be deployed on many of the currently known structured P2P overlays. The framework provides a programming model in which processing logic may be implicitly distributed with universality and expressiveness, much like the MapReduce framework. The programming model can be distinguished into a data model and a processing model. We adopt a key-value data model with possible duplicated keys to represent the data elements. However, the data model is conceptually a graph-based model, i.e., data elements can be organized into a graph structure. Now, where the data is list-based, then the graph degenerates into a list. This facilitates the mapping from the data elements to the Cayley graphs which in turn can be mapped to the structured P2P overlays.

Like MapReduce, the Katana processing model hides the parallelism mechanism from the users. Instead, it provides two MapReduce-like functions:

`kata` and `ana`. However, unlike MapReduce, the `kata` and `ana` functions are independent from one another and are not required to be executed one after another. While `kata` jobs are used to perform aggregation of some sort over the data elements, `ana` jobs are used to build datasets based on the input data elements (i.e., to produce *data graphs* out of the input *data graph*). The execution essentially follows a post-order depth-first traversal of an arbitrary spanning tree of the data graph.

2.4 Big Data Projects

Our experience on managing data in the cloud has enabled us to participate in several large projects with substantial funding. The first, funded by the National Research Foundation of Singapore (NRF), focuses on exploiting cloud for large-scale data analytics in environmental monitoring and waste management in megacities [1]. This requires building a platform for scientists to manage and analyze large amount of sensor data collected from two cities (Singapore and Shanghai) in order to detect emergent pollutants and manage waste. Our initial effort is to develop LogBase, a scalable log-structured database system that adopts log-only storage to remove write bottleneck and to support fast system recovery [36]. In our current implementation, LogBase provides in-memory multi-version indexes and various primary and secondary log-based index to speed up retrieval of data from the log. In addition, LogBase supports transactions that bundle read and write operations spanning across multiple records.

The second project, also funded by NRF, aims to develop a comprehensive IT infrastructure for Big Data management, supporting data-intensive applications and analyses. Our epiC project has formed the basis for us to investigate various issues such as iterative computations that cannot be well supported by existing systems. At this moment, we are investigating check-pointing, recovery and concurrency issues in supporting iterative processing required for data analytics.

Finally, the third project comes under the Sensor-Enhanced Social Media (SeSaMe) Centre [3] jointly funded by Zhejiang University, NUS and Media Development Authority (MDA). The SeSaMe research center focuses on long-term research related to sensor-enhanced social media that enables linking of static and mobile cyber-physical environments over the Internet by the abstraction of sensing, processing, transport and presentation. The center will also facilitate the design of social media applications on cyber-physical systems through research advances that will transform the world by providing systems that respond more quickly. In this project, our goal is to leverage the Cloud techniques to efficiently manage and retrieve streaming data from sensors, mobile phones and other real-world data sources to support the analytical jobs of real world problem and a tool to visualize the results. We are building a new Cloud-based streaming engine to handle requests efficiently and reliably.

3. TSINGNUS: A LOCATION-BASED SERVICE SYSTEM TOWARDS LIVE CITY

The NUS-Tsinghua Extreme Search (NExT) Center [4], funded by the Media Development Authority (MDA) of Singapore, is a joint collaboration between the NUS and Tsinghua University to develop technologies towards a livable city. The program brings together researchers from different fields (multimedia, networks, databases) from the two universities to facilitate *extreme search* over large amount of real-time and dynamic data - social media (e.g, blogs, tweets, q&a forum), video, image, textual (documents) and structured data - beyond what is indexed in the web.

TsingNUS [6, 19] is a location-based service system that focuses on exploiting database technologies to support location-based services. TsingNUS goes beyond traditional location-aware applications that are based solely on user locations. Instead, TsingNUS aims to provide a more user-friendly location-aware search experience. First our *location-aware search-as-you-type* feature enables answers to be continuously returned and refined as users type in queries letter by letter [45]. For efficiency, we proposed the *prefix-region tree* (PR-tree), a tree-based index structure that organizes the dataspace into a hierarchy of spatial-textual regions such that (a) the spatial component of nodes nearer to the root are larger, and (b) the textual component of nodes nearer to the root are prefix of the textual component of descendant nodes.

Second, TsingNUS offers efficient mechanisms to process spatial-keyword queries for both AND semantics (where all keywords must appear in the retrieved content) and OR semantics (where some keywords appear in the retrieved content) [42]. Our newly developed scalable integrated inverted index, I^3, is an inverted index of *keyword cells*. A keyword cell denoted (keyword w, cell c) refers to a list of documents that contain w and the associated spatial locality of the documents fall in region c. We have used the Quadtree structure to hierarchically partition the data space into cells.

Third, TsingNUS incorporates continuous spatial-keyword search to efficiently support continuously

moving queries in a client-server system [15]. We have developed an effective model to represent the safe region of a moving top-k spatial-keyword query. Such a region bounds the space for which the user (and hence the query) may move while the answers remain valid.

We are extending our work to road networks (e.g., finding frequent routes [7]) and to support a wider variety of query types (e.g., nearest group queries [41]). We are also exploring how users' social networks can be tapped upon to support more sophisticated queries.

4. INTEGRATED MINING AND VISUALIZATION OF COMPLEX DATA

The drive to find gold nuggets in data has resulted in the explosion of discovery algorithms in the past decade. Many of these discovery algorithms focus on specific data type. However, with the advances of technology, many applications now involve records with attributes of diverse data types, ranging from categorical, to numerical, to time series, to trajectories.

Knowing the relationships among all the different types of data can aid in the understanding of a patient health condition. For example, suppose we have a frequent itemset Male, Smoker and an interval-based temporal pattern Headache Overlap HighBloodPressure. If these two patterns occur together, it may raise an alarm as studies have shown that a male smoker who experiences headache with elevated blood pressure has a high risk of having cardiovascular disease.

Handling datasets with such variety is a challenge as the complexity of the problem can quickly grow out of hand. We have developed a framework to perform the integrated mining of big data with diverse data types [28]. The framework consists of algorithms for mining patterns from interval-based events [27], lag patterns involving motifs in time series data [29], spatial interaction patterns [32, 31], duration-aware region rules and path rules for trajectories [30]. With this, we are able to capture the associations among different complex data types and demonstrate how these patterns can be used to improve the classification accuracy in various real world datasets.

We have also developed a tool, in cooperation with the Center for Infectious Diseases Epidemiology and Research at the Saw Swee Hock School of Public Health, to generate and highlight interesting patterns discovered from the different data types. This tool will also allow the visualization of event incidences, clusters and heat maps. Ongoing research aims to develop an interactive system for the visualization and analysis of trajectories.

5. QUERY REVERSE ENGINEERING

To help users with constructing queries and understanding query results, we have developed an approach, termed Query by Output (QBO), to reverse engineer queries given an input pair of database and query output. Given a database D and a result table T = Q(D), which is the output of some query Q on D, the goal of QBO is to construct candidate queries Q', referred to as instance-equivalent queries, such that the output of query Q' on database D is equal to Q(D).

We have applied QBO to improve database usability in two contexts. In the first scenario, QBO is used to help users better understand their query results by augmenting the result of a query Q (w.r.t. a database) with instance-equivalent queries that describe alternative characterizations of their query results [34]. As an example, suppose that a university physician issues a query to his clinic's database to find students who have been infected with a skin rash over the past week. Besides returning the query result, if the database system had also computed and returned an equivalence-instance query that revealed the additional information that all the students in the query result either had recently returned from an overseas trip to region X or are staying in the same dormitory as those students, then the physician could have been alerted about a potential skin rash outbreak in those dormitories. Thus, it is useful to augment a query's result with alternative characterizations of the query's result to provide additional insightful information.

In the second scenario, QBO is used to generate explanations for unexpected query results that have missing expected result tuples [33]. As an example, suppose that a manager issues a query to compute the annual sales figures for each of her regional sales agents and she is surprised to find that Alice's sales performance is lower than that of Bob's, which is inconsistent with her impression of their results. The manager could issue a follow-up "why-not" question to clarify why Alice's sales figure is not higher than that of Bob's. Using QBO, the database system could respond to this why-not question with an explanation in the form of an alternative query (e.g., compute total sales for each sales agent excluding the period when Alice was on sick leave) which would have returned an output result that is consistent with the manager's why-not question. Thus, providing a capability to explain why-not questions would be very useful to help users

understand their query results. We are currently implementating a query acquisition tool based on QBO that enables users to construct queries from examples of database and query result pairs.

6. DATA ANALYTICS

In addition to developing novel platforms for efficient data analytical processing, we are also looking at bringing human into the loop.

6.1 CrowdSourcing

We are developing a data analytics system that exploits crowdsourcing to manage complex tasks for which human can offer better (especially in terms of accuracy) alternative solutions. Our system, called Crowdsourcing Data Analytics System (CDAS), is designed to support deployment of crowdsourcing applications [23, 13]. In CDAS, a task is split into two parts - the computer-oriented tasks and human-oriented tasks. Crowdsourcing is employed to handle the human-oriented tasks. The results of the two tasks are then integrated. CDAS has a number of features that distinguish it from other crowdsourcing systems. First, CDAS has a quality-sensitive answering model that guides the crowdsourcing engine to process and monitor the human-oriented tasks. To reduce costs, the model employs a prediction model to estimate the number of workers required in order to achieve a certain level of accuracy. To ensure the quality of the estimation, historical information on reliability of workers is used. In fact, we also inject tasks for which answers are known in order to gauge the reliability of the workers. In addition, CDAS adopts a probabilistic approach (instead of the naive voting-based strategy) to verify the correctness of answers from workers. The idea of the scheme is to combine vote distribution of the current tasks and the historical accuracies and reliability of workers to determine the quality of the current answers by the workers. The intuition is to give higher weights to reliable workers.

Second, since workers complete their tasks asynchronously, CDAS supports "online aggregation", i.e., answers (with quality bounds) are continuously displayed and refined as responses from workers are received. This reduces the initial response time to end-users significantly.

We have demonstrated the effectiveness of CDAS in terms of both performance and ease of use in two different applications. A twitter sentiment analytics system has been developed on top of CDAS for analyzing the sentiments of movie goers. Another image tagging system has been built to facilitate image tagging of Flickr images. We have also exploited crowdsourcing in web table mapping and schema integration.

6.2 Collaborative Visual Analytics

In this research, we study how people can collaboratively achieve certain tasks by sharing their data and analytics results through the social network.

We have set up the *Internet Observatory* project [5] with the goals to monitor and analyze the dynamic user-generated contents on the Internet, and to provide a platform for users to share their findings. To provide context, we index these dynamic contents via Wikipedia, a well-established online encyclopedia which have entries for large number of entities and concepts [22, 21]. As an example, consider the Wikipedia entry for `Senkaku Island Dispute`. Besides visualizing the Wikepedia entry, our system also displays *dynamic* information (obtained from other sources) that are related to `Senkaku Island Dispute` including URLs, images, tag summarization, community view and geographical view. Currently, our system provides users with a set of social websites that they can choose to logon to in order to extract related information. This allows users to link/compare them to other information and opinions on the Internet. By doing so, the user is implicitly adding his/her private data into a public pool for general analysis.

We have also started the *ReadPeer* project which aims to promote reading as a large scale social activity by integrating ebooks and social networks to encourage more people to read and discuss about the materials they read. Our ReadPeer system allows users to make annotations on ebooks, research articles or any documents in PDF format. These annotations can be linked to various multimedia contents like blogs, videos, images, web links etc. and shared to friends in a social network.

Our approach to collaborative visual analytics involve reorganizing social media messages around a center of focus like Wikipedia articles or ebooks instead of putting these messages in a plain news feed. This allows users of common interest to come together to share their insights and analysis. Central to this is the design of visual interfaces that allow users to communicate and understand each other's perspectives. Moreover, these interactions generate databases that capture a lot of interesting semantics through linkages of social media messages into a rich information network. Visualizing such a rich information network is challenging [43, 44, 38].

7. ACKNOWLEDGEMENTS

Many of our research are done in collaboration with

international visitors. These include Divy Agrawal, Elisa Bertino, H.V. Jagadish, David Maier and Tamer Ozsu. We also thank our research fellows for their contributions to our group. Finally, special thanks to our many graduate students - without them, we would not be where we are today!

8. REFERENCES

[1] Energy and environmental sustainability solutions for megacities. http://www.nus.edu.sg/neri/E2S2.html, 2013.

[2] epiC@NUS. http://www.comp.nus.edu.sg/~epic, 2013.

[3] SeSaMe. http://sesame.comp.nus.edu.sg/, 2013.

[4] The NExT Center. http://next.comp.nus.edu.sg, 2013.

[5] Trendspedia. http://www.trendspedia.com/, 2013.

[6] TsingNUS. http://tsingnus.comp.nus.edu.sg, 2013.

[7] H. Aung, L. Guo, and K. L. Tan. Mining sub-trajectory cliques to find frequent routes. In *SSTD*, 2013.

[8] Y. Cao, C. Chen, F. Guo, D. Jiang, Y. Lin, B. C. Ooi, H. T. Vo, S. Wu, and Q. Xu. Es^2: A cloud data storage system for supporting both oltp and olap. In *ICDE*, pages 291–302, 2011.

[9] G. Chen, K. Chen, D. Jiang, B. C. Ooi, L. Shi, H. T. Vo, and S. Wu. E3: an elastic execution engine for scalable data processing. *JIP*, 20(1):65–76, 2012.

[10] G. Chen, T. Hu, D. Jiang, P. Lu, K. L. Tan, H. T. Vo, and S. Wu. Bestpeer++: A peer-to-peer based large-scale data processing platform. In *TKDE (Special Issue for Best Papers in ICDE'2012)*.

[11] G. Chen, T. Hu, D. Jiang, P. Lu, K. L. Tan, H. T. Vo, and S. Wu. Bestpeer++: A peer-to-peer based large-scale data processing platform. In *ICDE*, pages 582–593, 2012.

[12] G. Chen, H. T. Vo, S. Wu, B. C. Ooi, and M. T. Özsu. A framework for supporting dbms-like indexes in the cloud. *PVLDB*, 4(11):702–713, 2011.

[13] J. Gao, X. Liu, B. C. Ooi, H. Wang, and G. Chen. An online cost sensitive decision-making method in crowdsourcing systems. In *SIGMOD Conference*, 2013.

[14] W. X. Goh and K. L. Tan. Katana: Generalized data processing on peer-to-peer overlays. In *IC2E*, 2013.

[15] W. Huang, G. Li, K. L. Tan, and J. Feng. Efficient safe-region construction for moving top-k spatial keyword queries. In *CIKM*, pages 932–941, 2012.

[16] H. V. Jagadish, B. C. Ooi, and Q. H. Vu. Baton: A balanced tree structure for peer-to-peer networks. In *VLDB*, pages 661–672, 2005.

[17] D. Jiang, B. C. Ooi, L. Shi, and S. Wu. The performance of mapreduce: An in-depth study. *PVLDB*, 3(1):472–483, 2010.

[18] D. Jiang, A. K. H. Tung, and G. Chen. Map-join-reduce: Toward scalable and efficient data analysis on large clusters. *IEEE Trans. Knowl. Data Eng.*, 23(9):1299–1311, 2011.

[19] G. Li, N. Zhang, R. Zhong, W. Huang, K. L. Tan, J. Feng, and L. Zhou. TsingNUS: A location-based service system towards live city (demo). In *SIGMOD*, 2013.

[20] Y. Lin, D. Agrawal, C. Chen, B. C. Ooi, and S. Wu. Llama: leveraging columnar storage for scalable join processing in the mapreduce framework. In *SIGMOD Conference*, pages 961–972, 2011.

[21] C. Liu, B. Cui, and A. K. H. Tung. Integrating web 2.0 resources by wikipedia. In *ACM Multimedia*, pages 707–710, 2010.

[22] C. Liu, S. Wu, S. Jiang, and A. K. H. Tung. Cross domain search by exploiting wikipedia. In *ICDE*, pages 546–557, 2012.

[23] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang. Cdas: A crowdsourcing data analytics system. *PVLDB*, 5(10):1040–1051, 2012.

[24] P. Lu, S. Wu, L. Shou, and K. L. Tan. An efficient and compact indexing scheme for large-scale data store. In *ICDE*, 2013.

[25] W. Lu, Y. Shen, S. Chen, and B. C. Ooi. Efficient processing of k nearest neighbor joins using mapreduce. *PVLDB*, 5(10):1016–1027, 2012.

[26] W. S. Ng, B. C. Ooi, K. L. Tan, and A. Zhou. Peerdb: A p2p-based system for distributed data sharing. In *ICDE*, pages 633–644, 2003.

[27] D. Patel, W. Hsu, and M. L. Lee. Mining relationships among interval-based events for classification. In *SIGMOD Conference*, pages 393–404, 2008.

[28] D. Patel, W. Hsu, and M. L. Lee. Integrating frequent pattern mining from multiple data domains for classification. In *ICDE*, pages 1001–1012, 2012.

[29] D. Patel, W. Hsu, M. L. Lee, and S. Parthasarathy. Lag patterns in time series databases. In *DEXA (2)*, pages 209–224, 2010.

[30] D. Patel, C. Sheng, W. Hsu, and M. L. Lee. Incorporating duration information for trajectory classification. In *ICDE*, pages 1132–1143, 2012.

[31] C. Sheng, W. Hsu, M. L. Lee, and A. K. H. Tung. Discovering spatial interaction patterns. In *DASFAA*, pages 95–109, 2008.

[32] C. Sheng, Y. Zheng, W. Hsu, M. L. Lee, and X. Xie. Answering top-k similar region queries. In *DASFAA (1)*, pages 186–201, 2010.

[33] Q. T. Tran and C. Y. Chan. How to conquer why-not questions. In *SIGMOD Conference*, pages 15–26, 2010.

[34] Q. T. Tran, C. Y. Chan, and S. Parthasarathy. Query by output. In *SIGMOD Conference*, pages 535–548, 2009.

[35] H. T. Vo, C. Chen, and B. C. Ooi. Towards elastic transactional cloud storage with range query support. *PVLDB*, 3(1):506–517, 2010.

[36] H. T. Vo, S. Wang, D. Agrawal, G. Chen, and B. C. Ooi. Logbase: A scalable log-structured database system in the cloud. *PVLDB*, 5(10):1004–1015, 2012.

[37] J. Wang, S. Wu, H. Gao, J. Li, and B. C. Ooi. Indexing multi-dimensional data in a cloud system. In *SIGMOD Conference*, pages 591–602, 2010.

[38] N. Wang, S. Parthasarathy, K. Tan, and A. K. H. Tung. Csv: visualizing and mining cohesive subgraphs. In *SIGMOD Conference*, pages 445–458, 2008.

[39] S. Wu, D. Jiang, B. C. Ooi, and K. L. Wu. Efficient b-tree based indexing for cloud data processing. *PVLDB*, 3(1):1207–1218, 2010.

[40] S. Wu, F. Li, S. Mehrotra, and B. C. Ooi. Query optimization for massively parallel data processing. In *ACM SOCC*, 2011.

[41] D. Zhang, C. Y. Chan, and K. L. Tan. Nearest group queries. In *SSDBM*, 2013.

[42] D. Zhang, K. L. Tan, and A. K. H. Tung. Scalable top-k spatial keyword search. In *EDBT*, pages 359–370, 2013.

[43] F. Zhao, G. Das, K. Tan, and A. K. H. Tung. Call to order: a hierarchical browsing approach to eliciting users' preference. In *SIGMOD Conference*, pages 27–38, 2010.

[44] F. Zhao and A. K. H. Tung. Large scale cohesive subgraphs discovery for social network visual analysis. *PVLDB*, 6(2), 2012.

[45] R. Zhong, J. Fan, G. Li, K. L. Tan, and L. Zhou. Location-aware instant search. In *CIKM*, 2012.

What does an Associate Editor actually do?

Graham Cormode
G.Cormode@Warwick.ac.uk

ABSTRACT

What does a Associate Editor (AE) of a journal actually do? The answer may be far from obvious. This article describes the steps that one AE follows in handling a submission. The aim is to shed light on the process, for the benefit of authors, reviewers, and other AEs.

1. INTRODUCTION

Journal publications are an important part of the propagation of results and ideas in computer science. Papers in prestigious journals reflect well on their authors, and serve to provide a full, detailed and peer-reviewed description of their research. Yet, the process from submission to decision is opaque. A researcher typically submits their paper to a journal and then waits months (sometimes many months) before receiving a set of reviews and a decision on whether the journal will pursue publication of the submission. It is far from obvious to the researcher exactly what is going on during this time.

The purpose of this article is to shed more light on this process, by describing the typical sequence of events from the perspective of the associate editor. The hope is that this serves multiple purposes:

- To help authors understand the process, and allow them to make their submissions with this knowledge.
- To help journal reviewers understand their role in the process, and how they can be most effective in helping to determine the right outcome for a submission.
- To help me (and, by extension, other associate editors) think of the process more clearly, and optimize our role within it.

The editorial structure of a journal varies between titles, but in general there is an editorial board which consists of an Editor-in-Chief (EiC) and multiple Associate Editors (AE). The role of this board is to determine which papers to accept for publication in the journal.

In general, the EiC receives new submissions and allocates these to AEs for handling through the review and decision process. The complete range of tasks performed by the EiC is not necessarily known to the AE: there are many "behind-the-scenes" tasks performed that they do not get to see[1].

This article focuses on the role of the AE in the editorial process, in order to answer the question "What does an Associate Editor actually do?". The answer is far from obvious: for example, one thing the AE does *not* typically do is "edit" papers in the popular sense of the word[2]. Rather, the AE's main task is to make editorial recommendations to the EiC about what decision should be made on submitted papers.

To accomplish this, the AE has a seemingly simple set of responsibilities: to obtain referee reports for each paper they are assigned, and use these to make their recommendation for the paper, in a timely fashion. The execution of these tasks however requires quite a substantial amount of effort; moreover, this effort is concentrated in areas that might not be initially obvious. To explain this, I will describe the detailed sequence of steps that I follow between receiving a new assignment and providing my recommendation. A standard caveat applies: this description reflects my perspective and processes, informed by input from others (for example, [5]). Different AEs will no doubt have different approaches to the job. The author takes no responsibility for any loss, damage, or injury that may result from following any advice in this article.

Outline. In Sections 2 and 3, I outline the two main components of the AE's job: initial handling and selection of reviewers for a paper (Section 2), and obtaining a decision for a paper (Section 3). In Section 4, I offer some suggestions for reviewers, authors, and associate editors in turn.

[1]In more blunt terms, I don't fully know what the EiC does.
[2]The person who does make edits to accepted papers is the sub-editor, although in my experience this primarily involves the insertion or removal of commas.

2. SECURING REVIEWERS

Step 0: pre-processing. When a paper is submitted to a journal, it receives some attention before being assigned to an AE for handling. The EiC, and possibly an editorial assistant, will look over the paper. The general goal of this step is to check that the paper is suitable for further processing: Does it meet the formatting requirements? Is it generally on-topic for the journal? does it have a clear, novel technical contribution? Is it possible to open the files? Is it written in the language used by the journal? If the paper passes these checks, then the EiC will identify an AE to handle the paper, and assign it to them. The choice of which AE will handle the paper may depend on many factors: whether it falls within the AE's area of expertise, the relative workload of the AEs, avoiding potential conflicts of interest between the authors and the AE, and so on[3].

In most journals, the paper is handled via a web-based manuscript system (with a generic sounding name like *ScholarCentral* or *ManuscriptOne*), which tends to enforce a particular workflow. The web-based manuscript system (WBMSS) will generate email alerts to each participant when they have a task to perform. So when a paper is assigned to me, the WBMSS will generate an email message telling me that I have work to do.

You've got email. My process on receiving a new paper to handle is as follows: I first sigh[4], realizing that this means more work to do. Then I am overcome with excitement about the prospect of guiding a fresh paper through the journal submission process.

I next take a print out of the main paper and any cover letter. As soon as possible, I run a hot bath, and immerse myself in the water and in the paper[5]. I then read the paper to get an idea of what it is about, roughly what techniques is it using, and what papers are most relevant to the work in hand.

My objective in this phase of the process is to identify a set of researchers to contact and ask them to provide a review of the submission. As such, my approach is quite different to when I am reviewing a paper myself. As an AE, I do not find it necessary to comprehend every last detail of the paper, or even to grasp all of the ideas presented. Rather, my goal is to find experts who can understand the paper in detail, and provide commentary on its significance and novelty. Consequently, I try to avoid forming a strong opinion about whether the submission should be accepted: the bulk of that work will be on the reviewers. However, based on my initial reading of the paper, I will have a sense of the general level of the paper.

Sometimes it is clear that the paper does not meet the standards of the journal. In such cases, an AE may provide an "administrative reject" decision (also known as a "desk reject"). I do this when I am certain that the paper stands almost no chance of eventually being accepted. In particular, I want to be able to provide the authors with a supportable reason for the reject decision and feedback that they can make use of. Reasons I consider suitable to motivate an administrative reject include if the submission is presented so badly it is impossible to understand any of what is being said; if the results very clearly duplicate prior work; if the topic of the paper seems very much out of scope for the journal; or if the submission includes text that appear in other previously published papers and thus violates the journal's plagiarism policy. In my experience, submissions meeting any of these criteria are not common, perhaps because the EiC catches them before they are assigned to an AE.

There are still some papers which I believe are borderline for the journal, but which do not match any of the above conditions. In these cases, I can invite reviewers to review the paper, even though I think its prospects are poor. It is better to allow a seemingly poor paper a fair chance with expert reviewers, than for an AE who is not an expert in its area to deny it any chance. This gives the authors of the paper a fuller set of reviews, which is hopefully of use to them. The tradeoff is that I am asking reviewers to give their time to review what may be a poor paper. My rationale is that reviewing is part of the service we owe the community in return for submitting our own papers, and we cannot always expect high-quality papers to read. Moreover, it should be a relatively quick task for an expert if the submission is indeed of low quality to make an assessment and to prepare a short review highlighting the deficiencies. I can invite fewer reviewers (say, two), if I think that there is a good chance that they will both provide negative reviews.

As a third option, I sometimes desk reject based on a fixable issue, such as problems with figures or formatting. In the feedback to authors, I let them know that it is permissible to resubmit a corrected version of the paper. I also indicate that I believe that such a revision is unlikely to meet the high standards of the journal. This leaves the door open for the authors to resubmit, while indicating heavily that they would do well to reconsider their choice of venue.

Picking Reviewers. After getting a sense of the paper, my next step is to identify a set of potential reviewers to invite. I think about the paper as I understand it, and

[3]I suspect that a whole new article could be written about the job of the EiC, and I would encourage someone to do so.

[4]Or, according to taste, shriek, cry out, rend my clothing, or ask "Why me?"

[5]People often ask me why I read papers in the bath. I patiently explain that it would be hopeless to try to do this in the shower.

which researchers are active in that area or related areas. I cast my mind over papers I have read, presentations I have seen, and conversations I have had to identify who is suitably expert on the topic. There doesn't have to be an exact match – perhaps the application is unusual, but a reviewer has used similar techniques.

I also draw ideas for reviewers from the paper. Does the paper make extensive reference to some prior work? Does it compare to a method described in a previous paper? Then there is a good chance that I will invite the authors of these papers (assuming that they do not overlap with the authors of the current submission) to perform the review. I may do some speculative searching – are there keywords or problem descriptions from the paper that I can find other papers about online? In particular, can I find papers on similar topics published in the same journal – since I feel the authors of those works owe a review back to the journal.

After brainstorming for a while, I usually have a list of half a dozen potential reviewers. I do some additional research on them to ensure that they are well-placed to help. Before inviting each reviewer, I check their homepage and their entry on DBLP. I look at the titles and venues of their papers, and years in which they have been active in this area, and also descriptions of their current role and activities.

Other commitments. I tend to avoid asking people who indicate that they are the head of a large research group, chair of their department and active in running a start-up at the same time. Such people tend to be too busy to perform reviewing tasks[6]. Advanced graduate students can be a good fit because they know their focus area very well, and have very few other pressing demands on their time[7]; however, it is sometimes hard to tell which students are mature enough in their area without a personal recommendation. So the bulk of reviewing falls upon faculty and researchers who don't appear too busy, or don't yet realize how busy they are.

I avoid asking EiCs and AEs of any journal to perform a review: they are usually far too occupied with the submissions for their own journal. In particular, I avoid asking an AE from the same journal to assist [8].

Still Active? The editor's curse is to find someone who has worked on some highly related topics, only to discover that their last publication was in 1999. Usually this means that they have left research for another career, retired, or abandoned this area of study[9]. In some cases, I identify a reviewer who would be perfect to help with a paper, only to discover that they are no longer alive, which I find most inconsiderate.

Following this analysis of reviewers, I pick a shortlist of 3 or 4, and start to send out invitations. The WBMSS typically has a default invitation template describing the expectations. I personalize this invitation, to give some indication of why I have invited the reviewer: for example, because I think the submission relates to their expertise on a topic, or because it compares to their system, for example. My hope is that this personal touch will make them more likely to accept the invitation. The invitation can also indicate if the paper is a resubmission, an invited submission or an extended version of a conference paper.

I might include the submitted manuscript with the invitation. When I am invited to review, I often find it helpful to quickly scan the submission, to determine how relevant it is and much effort it will be. When suitable, I like to give other reviewers this opportunity. However, I must admit, when a paper seems particularly long and technically dense, I may avoid sending it, for fear of scaring off the potential reviewer.

Dealing with rejection. Inevitably, some invitations to review will be met with rejection. Indeed, in my experience about half of responses are negative. This can be for many reasons, of varying validity: the invitee is too busy, does not consider themself an expert on the subject matter, does not find the paper interesting, or just doesn't feel like it on the day. A negative response does not annoy me (unless I feel that the paper really was spot-on for the reviewer). What does irk me are two things:

Tardiness – it should not take a long time to respond to a review request. If people are actively at work, I would hope to hear a reply within a couple of days; if traveling or otherwise tied-up, I would still hope to hear within a week or so[10]. It pains me when an invitee sits on a request for weeks, and then declines (possibly only after a reminder). Even when the invitation is accepted after a long pause, this can be a troubling sign, as it indicates that the review itself may be similarly delayed.

Lack of alternative suggestions – my favourite type of response is actually a very fast negative response that comes with a list of suggested alternate reviewers. This means that the invitee has thought about the invitation, understands that they are unable to commit to it, but has

[6]They often appear to be too busy even to respond to review requests.

[7]Graduation can wait.

[8]I hope they realize that this is why I turn down their corresponding review requests. Ideally, the EiC would always assign the paper to the most expert AE on that topic. However, I have gradually come to realize that EiCs are less omniscient than one might at first imagine them to be.

[9]One does not like to name names, but on multiple occasions I have had papers which refer heavily to the work of S. Brin and L. Page. However, these two stopped publishing in the 1990's, and have not responded to any of my requests for reviewing. I can only assume that these promising researchers have given up on academia, and followed a less rewarding career in industry.

[10]Everyone checks their email while on vacation, right?

considered it enough to come up with a list of others who may be able to help. This is particularly valuable when the area of the submission is less familiar to me. As a reviewer, I suggest alternates when I am unable to assist – unless I really don't know the topic. As a result, when an subject matter expert declines, I often follow up with an email pushing for some suggested alternate reviewers. I encourage people to feel obliged to provide alternates when declining an invitation.

Adding more reviewers. When reviewers decline an invitation, I need to find more reviewers to invite. Sometimes I have some back-ups already picked, or can take advantage of suggestions from those who have declined. I avoid having more than four "active" invitations at one time, in case all reviewers accept: it is redundant to have a large number for one paper. Often though, I need to find some new candidates. This is perhaps the toughest part of the job, as it means further head scratching to come up with good candidates. It is quite dispiriting when a large number of reviewers have declined to review a paper. The worst case is when the paper is quite specialized, and all the natural candidates have been tapped. It is particularly galling when, after prompting for other reviewers, the suggestions consist of candidates who have already declined. At this stage, the AE can feel that the task of finding enough suitable experts to evaluate a paper may be impossible. However, with persistence, enough reviewers will eventually agree.

Reaching Acceptance. When sufficient reviewers have agreed to review a paper (usually three or four), and dates for the review have been agreed, the initial phase of the process is complete. I can sit back, relax, and wait for the reviews to arrive.

3. GETTING TO A DECISION

The whooshing sound they make as they fly by. When I first started working as an AE, I imagined that the bulk of the effort was in weighing up the reviews for a paper, and synthesizing these to come up with a careful, considered decision and rationale for it. This a much less significant part of my work than I had thought. Indeed, it seems that much of the effort of the AE is in reminding, cajoling and threatening reviewers who have agreed to provide a review, but who fail to fulfil their promises.

In the ideal situation, reviewers will perform their task within the allotted time (typically, six weeks to a few months), and deliver a carefully thought-out, clearly expressed review. Indeed, most reviewers do an excellent job in this regard, and I am truly grateful to them. However, there are many cases where things do not follow this outline, and more active involvement is required.

The WBMSS usually includes a "due date" for each review (which can be set by the AE), and may automatically remind the reviewer as the deadline approaches and is passed. In addition, around the time of the deadline, I send a personalized reminder, as this is harder to ignore than an automated message. I do not keep detailed statistics, but while many reviews are received on time, it is a sad fact that a large fraction are late. A little tardiness is forgivable, but after more than a week, it starts to become a problem. Many journals strive to have a rapid turnaround time for submissions, and delayed reviews are the biggest obstacle to achieving this goal [4, 5].

Checking this requires more of my attention. I have to keep an eye on which reviews are late, and send reminders to reviewers, requesting that they make good on their promise, and deliver their review. The pressures that I can bring to bear are limited: I can send increasingly plaintive requests, or express my displeasure or anguish at the continued delay; I can try to provoke guilt or regret in the reviewer; but there are few direct actions I can take against the tardy reviewer. Persistence is my only weapon. In a few cases I have given up on receiving a review when the other reviews received were sufficient to reach a decision.

The reviews are in. When I do receive a review, I read it carefully, and check that there are not any obvious problems with it. Problems in reviews are rare, but occasionally it may be clear that the reviewer's standards are not calibrated for the venue (too harsh, or too lenient); or that the recommendation does not align with the content of review (e.g. many major flaws highlighted, but an "accept" recommendation). Reviews can sometimes be improved by clarifying what is expected from a revision, and ensuring that the discussion is as objective as possible. The AE can ask a reviewer to revise or elaborate their review. Very rarely, there may be inconsistencies across reviews that are resolved by an (email) discussion with the AE in the middle.

The Big Decision. When there are sufficient reviews for a paper, I can make a decision. The typical number is three, but more or fewer is possible. I am happy to recommend rejection for a paper on the basis of two reviews which agree on this outcome, or even one in extreme cases. For a positive recommendation, I prefer to have received three reviews, even if they are not unanimous. Collecting four reviews is reasonable (and acts as insurance against one reviewer going awol); more than four is unusual except for very selective journals.

I usually find it fairly swift to make a decision: reviews often agree on the general level of quality and interest in a submission. Some normalization is needed based on the standards of the journal, but in general it is quick to weigh the comments and scores of the re-

viewers, and reach a consensus. The process is guided almost exclusively by the reviews–my opinions of the paper carry almost no weight at this point[11]. The first decision is a binary one: Is there any prospect of publishing this paper in the journal? Does it show enough potential and interest? If not, then the recommendation is to "reject" the paper. This recommendation is accompanied by a justification, summarizing the reasons for rejection: I identify the main reasons from reviews that led to the decision. It may include more or less encouragement to submit to another venue, especially if the submission was ultimately judged out of scope or below threshold for my journal. The authors may appeal a reject decision, either to the AE or the EiC, but without evidence of serious unfairness this is unlikely to alter the outcome. A rejected paper is sometimes resubmitted to the same journal, after some revisions. Most journals will try to catch this, and either reject automatically, or assign it to the same AE to handle.

If the paper is not rejected, there are three possible recommendations: "accept (as is)", "minor revision", and "major revision". The exact semantics of this vary depending on the journal, but as a rough guide, a major revision will be returned to the same reviewers to get their opinion on the new version; a minor revision will be scrutinized by the AE; and an accept will move straight into the publication queue. However, the AE has a lot of leeway: a minor revision may be sent out to reviewers; and a major revision may be sent only to a subset of reviewers, or new reviewers may be added. I won't spell out all the situations that can arise, but the underlying issue is the same: before giving an "accept (as is)" decision, I want to be certain that the paper represents a sufficient contribution for publication in the journal. When the reviews indicate some notable questions or concerns, I want to be assured that these are suitably addressed before recommending the paper for publication. Sometimes I can do this myself (based on the revised submission, and any cover letter or list of revisions, and comparing these to the original reviews); or I may seek the opinion of the original reviewers on such questions.

Recommendations and Decisions. You may notice that an AE makes a "recommendation", not a "decision". This is deliberate terminology: it is the EiC who makes the decision, not the AE, who merely recommends an outcome. I will let you into a secret: I have not encountered cases where the EiC's decision did not follow the recommendation of the AE, although this does happen. I find that this is a useful way of thinking about the process. It reminds me that I have to justify my recommendation both to the authors and to the EiC; I am not making decisions at my whim. Once I submit my recommendation on a paper to the EiC, I can again sit back: my work – for now – is done.

Revisions. For revisions, the process starts over again – selecting reviewers, obtaining reviews, and making a recommendation. Typically, one invites the same set of reviewers, although there is the option to add new reviewers (if additional input is needed), or drop some (for example, if they were entirely satisfied with the previous version). There can be multiple rounds of revision, but if major issues remain after a first revision, it is common to move towards a reject. Once a reject or accept is reached, the AE's involvement with the paper is concluded.

4. RECOMMENDATIONS

Based on this description of the process, I have a number of recommendations and requests for those involved in the journal review process:

4.1 Recommendations to authors.

It is easy to imagine that a journal will immediately recognize the novelty and importance of a submitted paper, and that the editors will quickly identify experts who can judge the merits of the submission. However, the reality is perhaps less ideal: there is no guarantee that the EiC will be able to match the paper to the best AE for the paper, or that the assigned AE will be able to identify and secure the most expert reviewers. Authors can help this process along:

Suggest suitable Associate Editors. It is often appropriate to suggest an AE to handle the paper. Take a look at the editorial board, and see which AEs have familiarity with the area. The suggestion usually can be communicated to the EiC as part of the cover letter, or within the WBMSS.

Suggest suitable Reviewers. Before my experience as an AE, I did not think it was necessary to suggest reviewers: the journal staff should easily be able to identify an expert set of reviewers. Proffering suggestions seemed to imply that the nominees were my cronies. Now I realize that it is very valuable to suggest reviewers: there is no guarantee that the AE will be a leading expert in the domain of the paper, and I find that reviewer suggestions are useful input to me as an AE. I carefully evaluate suggested reviewers, and only follow up if it is clear that they are suited for the paper, and do not have conflicts of interest with the authors[12]. I tend to invite only one or two suggested reviewers, and fill out

[11] Occasionally, an AE may enter their own review for a paper they are handling on a topic are familiar with, especially if the invited reviewers have not done a timely job. Then this review is weighed up with the others.

[12] In particular, it is important to avoid inviting the authors to review their own paper, which is not unprecedented

the rest of the panel with "independent" reviewers, to avoid any issue of bias. Authors should realize that their suggestions may not accept the invitation, and there is little value in suggesting a "big-name" researcher who is too busy. Lastly, some journals also allow authors to indicate "non-preferred" reviewers. I can think of few situations where this is of use to authors, and it seems that there should be some clearly articulated explanation.

Think about your citations. Think carefully about which works you cite, and whether there are any important references missing. An AE will often look to the bibliography for potential reviewers to invite. So authors should realize that their bibliography is another list of "suggested reviewers". They should also reflect on how fairly they describe and compare to prior work, since the authors of those works may be called upon to judge the submission.

Optimize your revisions. As noted above, the revision will be handled by the same AE as the original submission, and will typically be read by the same reviewers. It is therefore sensible to optimize the revision accordingly. Make a cover document containing each review, and indicate how you respond to each point: what changes were made, and where. It is OK to disagree with a reviewer comment, so long as you explain why. It is also helpful to indicate which sections have changed in the paper, via highlighting[13]. This takes extra work, but this type of effort can make the review process go much more smoothly, and hence speed the paper to publication.

4.2 Recommendations to Reviewers.

These are perhaps less recommendations than pleas:

Respond swiftly and decisively to requests. As an AE, my goal is to provide well-informed decisions to authors in a timely fashion. This starts with responding to the initial review request. Please don't sit on a review request for weeks: it is usually only the work of a few moments to determine one's current level of commitments, and availability to accept a new task. As noted above, a swift response is often appreciated, even if it is negative. Please also provide alternate reviewer suggestions as a matter of course. Often, I receive a request and I think "Why are they asking me? Why don't they ask X?". The reason may be that the AE does not know that X is the expert on this topic – so please inform them of this! You can also use declining a review request as an opportunity to advance the career of a more junior member of your community, by suggesting someone less well-known.

Honour your commitments. When you accept to perform a review, you are making a commitment to deliver the review by the date agreed. This commitment should be taken seriously. It is easy to devalue the importance of review work – after all, it is "voluntary" work. However, I view reviewing as an obligation: when we submit papers, we expect them to receive appropriate and timely reviews, and so we should perform reviews similarly. It is tempting to think of reviews as less important than the many other demands on our time, (our own research, teaching, and funding deadlines) and allow the review to get progressively later and later. But this is quite unprofessional. It delays the process for authors, who need to get timely decisions in order to publish their work and progress their careers.

It goes without saying that you should do a good, careful job in reviewing the paper. For guidance on this, there are several good articles on the topic [3, 6, 1][14].

You should *always* accept a request to review a revision of a paper. The work involved should be much less than to perform an initial review (especially if the authors have suitably optimized their revision). If you asked for changes, then you should at least look at the response.

Accept a reasonable number of requests. It is hard to load-balance incoming review requests: sometimes, many arrive in close proximity. However, as indicated above, it is important to be an active participant in the review process, and do your fair share. One heuristic is to perform $3-4$ reviews for each submission you make (assuming that each paper does have multiple authors), but more senior people may need to do more.

Be aware that a journal review brings different expectations to a conference review. A journal review is expected to be in greater depth, and to more carefully scrutinize the whole paper. Consequently, the review should attempt to evaluate the paper in full, or be explicit about which sections could not be verified. Journal papers may also be (much) longer than a typical conference submission, so one to several months is allotted to perform the review – do not interpret this as permission to leave the review to the last minute.

4.3 Recommendations to Associate Editors.

The above discussion has outlined the workflow I tend to follow in handling a paper. Implicit in this are several recommendations and considerations:

Be considerate of authors. Your goal as an AE is to oversee a fair and timely handling of submissions to your journal. So try to ensure that each submission has

`http://barcorefblog.blogspot.com/2012/10/fake-peer-reviews.html`

[13]This has the advantage that it will focus the attention of the reviewers on just those parts of the paper; otherwise, they may re-read the whole paper, and come up with additional comments and things to change.

[14]As well as some that are laughable, e.g. [2].

a fair chance, by identifying and inviting suitable reviewers, and using these to make good decisions on papers. In some cases, the most considerate thing to do is to swiftly reject a paper, rather than enter it into a lengthy review process, taking up reviewers' effort, and ultimately reaching the same outcome.

Be considerate of reviewers. Try to identify reviewers who are suited to the paper, and try to avoid asking the same reviewers to help with a lot of papers. Be understanding when reviewers need more time to review a paper, while firmly reminding them of their obligation. Remember that reviewing is a mark of service to the community, and an indication of the esteem with which the opinion of the reviewer is held, so be sure to allow junior researchers the opportunity to participate in the review process. This can also be a learning opportunity for them to see firsthand how peer review works in practice, and to calibrate their opinions against the reviews of others.

Be considerate of yourself. When I started as an AE, I had high aspirations: I would read each paper in detail, and provide my own review and comments in addition to those of the invited reviewers. This lasted for exactly one paper. For journals with high throughput, you may handle 20-30 papers per year, on a wide variety of topics, and it simply is not practical, nor a good use of your time, to try to do too much. Stick to the core tasks, and you will be doing the community a service.

By way of guidance, here are my estimated times for handling a submission. Of course, these can vary: an obviously unsuitable paper may be faster to handle.

Read and think about paper: 1-2 hours

Search for and invite initial reviewers: 1 hour

Handle review responses, and find replacement reviewers: 1-2 hours.

Receive and process reviews: 0.5 hours total

Chasing reviewers to deliver their reviews: 1 hour

Re-visit paper, and formulate recommendation: 1 hour

5. CONCLUDING REMARKS

This is the end of what I have to say.

Acknowledgments. I thank Jian Pei for many helpful comments and suggestions.

6. REFERENCES

[1] Mark Allman. Thoughts on reviewing. *ACM SIGCOMM Computer Communication Review (CCR)*, 38(2), April 2008.

[2] Graham Cormode. How not to review a paper: The tools and techniques of the adversarial reviewer. *SIGMOD Record*, 37(4):100–104, December 2008.

[3] Ian Parberry. A guide for new referees in theoretical computer science. *Information and Computation*, 112(1):96–116, 1994.

[4] Richard Snodgrass. CMM and TODS. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 34(3):114–117, September 2005.

[5] Richard T. Snodgrass. ACM TODS associate editor manual. `http://tods.acm.org/editors/manualFeb2007.pdf`, January 2007.

[6] Toby Walsh. How to write a review. `http://www.cse.unsw.edu.au/~tw/review.ppt`, 2001.

Report on the first Workshop on Innovative Querying of Streams

Michael Benedikt
University of Oxford, UK
michael.benedikt@cs.ox.ac.uk

Dan Olteanu
University of Oxford, UK
dan.olteanu@cs.ox.ac.uk

1. INTRODUCTION

The first workshop on

INnovative QUErying of STreams (INQUEST)

was held on September 25-27, 2012 in the Department of Computer Science of the University of Oxford (UK). It was sponsored by the UK's Engineering and Physical Sciences Research Council (EPSRC), as part of the project "Enforcement of Constraints on XML Streams".

Stream processing represents a thriving area of research across the algorithms, databases, networking, programming languages, and systems research communities. Within the database community, a "classical" problem is query processing on streams of discrete tuple-oriented data. One goal of the workshop considers the way recent developments add complexity to this problem:

- how does the setting change when data to be considered by queries is not relational, but has nested structure, such as XML or JSON?
- conversely, how does the setting change when data to be considered consists of RDF triples?
- how does the presence of noise in the data impact query processing?
- how does stream processing change when querying requires not only access to the data, but reference to external knowledge, which can also be changing?
- how does processing change in a large-scale decentralized setting?
- what new demands on stream query processing arise from social media applications? Is it only the processing architecture that changes, or do the queries change as well?

In addition to looking at new developments in stream processing, the workshop aimed to bring together researchers with different perspectives on the topic. We solicited and received participation from researchers working primarily on stream architectures and systems as well as those working on stream algorithms; the participants included researchers working on the computation of particular aggregates in streaming fashion as well as those looking at high-level languages for describing queries.

The workshop was by invitation only. There were 52 registered participants, ranging over 20 institutions. The formal part of the workshop program consisted of 19 invited lectures, grouped by topic.

In what follows, we present the main ideas and issues proposed by the speakers. Finally, discussions arisen during the workshop and concluding remarks are presented. The slides of workshop talks can be found on the current workshop web page:
`http://www.cs.ox.ac.uk/dan.olteanu/inquest12/pmwiki.php`

2. STREAMING OF SOCIAL DATA

This session covered challenges in building scalable infrastructure for managing social media streams and in extracting valuable information from social media streams such as emergent topics.

Sebastian Michel considered the problem of emergent topics discovery by continuously monitoring correlations between pairs of tags (or social annotations) to identify major shifts in correlations of previously uncorrelated tags in Twitter streams [1, 2]. Such trends can be used as triggers for higher-level information retrieval tasks, expressed through queries across various information sources.

Mila Hardt gave two talks on aspects related to managing streams at Twitter, in particular on infrastructure to enable processing of 400 million tweets a day and real-time top queries. Mila explained how stream processing needs at Twitter eventually led to the development of the open-source projects Storm and Trident[1] for large-scale high-performance distributed stream processing. She also pointed out

[1]https://github.com/nathanmarz/storm

current challenges at Twitter in providing support for fault tolerance, online machine learning by trading off exploration and exploitation, and approximating aggregates (such as counts). An interesting exercise involving the audience was on thinking how topic ranking is done at Twitter.

Daniel Preotiuc-Pietro introduced the Trendminer[2] system for real time analysis of social media streams [19]. Trendminer's scalability relies on the MapReduce framework for distributed computing. Daniel also presented how to build regression models of trends in streaming data using TrendMiner [21].

3. STREAMING AND THE SEMANTIC WEB

Stream processing has emerged as an important challenge in the new field of managing linked and semantic data. The workshop featured three talks on efforts in managing streams of linked data: one by Emmanuel Della Valle, covering work done in Politecnico Milano, one by Manfred Hauswirth, covering work done at *DERI* on platforms for linked data stream, and by Darko Anicic, covering joint work with Sebastian Rudolph and others at Karlsruhe Institute of Technology.

The requirements of a stream processing system for semantic data include support for "continuous querying" – queries that remain in place, with answers evolving as new data arises – and support for reasoning with external knowledge. The approach presented in Della Valle's talk involves merging the approach used for relational continuous query language with SPARQL. The resulting language, *C-SPARQL* [4], allows one to filter from a stream, using continuous-query window commands to control the sampling method, but SPARQL graph patterns can now be used within the filters.

Anicic outlined a different language approach. The *ETALIS* system [3] supports stream reasoning by embedding both temporal relational rules within a logic programming formalism. To better support the standards suite of the semantic Web, *ETALIS* supports a proper extension of SPARQL for dealing with event-processing on streams, *EP-SPARQL*.

Of course, using stream processing on large-scale linked data involves more than just developing a language or even a query processing engine. Hauswirth's talk outlined the entire set of issues needed to build an application that integrates and processes sensor output using linked data. This includes a continuous query evaluation system specific to linked data, *CQELS* [18], but also addresses the modifications needed to storage, protocol, RESTFul services, date interchange formats, and data integration technology needed to exploit these query languages in real-world applications.

[2]https://github.com/sinjax/trendminer

4. STREAM MONITORING

Monitoring of streams is a good example of a subarea of streaming where different communities define the objectives in radically different ways, and attack the problem using very different techniques. For the verification community, monitoring appears in the form of run-time verification – for example, continuously monitoring reactive systems for violation. The focus is normally on temporal constraints. Issues of space consumption are critical, as in most stream-processing applications, but there is also a need to integrate the constraint language and the monitoring engine with data structures maintained in the code being monitored. In databases both the constraint languages and the monitoring model are normally quite different; constraints naturally focus on properties of data values (e.g. as in classical dependencies), while monitoring occurs both in batch mode and in response to discrete updates. Both of these communities have dealt with monitoring as a component with a very well-demarcated set of functionality within a larger system. In contrast, monitoring data has a broad meaning within data-oriented applications, with integrity-constraint validation being only one aspect of it.

Felix Klaedtke's talk came from the perspective of run-time verification. He focused on online monitoring of integrity constraints, where the constraints deal with the evolution of data over time, and are thus expressed in a variant of first-order temporal logic. He explained both the system and a set of algorithms for efficiently monitoring these constraints [5]. In this work, ideas from runtime verification and the database community (particularly, temporal databases) interact.

Lukasz Golab looked at properties of streams of relational data, focusing on two natural set of constraints that deal with both temporal and more traditional relational aspects. He defined sequential constraints, which generalize functional dependencies to account for order, and conservation laws [12] that are specific to the context of pairs of numeric streams corresponding to related quantities. He presented methods for checking these constraints in off-line fashion, as well as methods for seeing the extent to which they are violated.

Mariano Consens talked about monitoring in the broader sense – how can the quality and the accesses to data records be monitored off-line in the presence of large volumes of linked data. His work focuses

on privacy issues in data, presenting an integrity language that allows one to formulate constraints expressing that a privacy violation has occurred. He also presented a system providing an end-to-end solution for auditing privacy constraints, including a means for integrating records from diverse datasources, for expressing privacy policies and constraints, and for detecting violations.

5. XML STREAMS

XML is notable for being a data model where very strong notions of streamability can be formalized for very expressive query and schema languages. Joachim Niehren looked at one natural formalization for node-selecting queries: the ability to determine at any point in an XML stream which nodes "must be" in the query result, where "must" means that they will be in the result in any possible extension. Niehren presented automata-theoretic methods of solving this "earliest answer problem", along with lower bounds.

While Niehren's talk focused on node-selecting languages such as XPath, Pavel LaBath looked at stream-processing of the World Wide Web consortium's XML transformation language, XSLT. He presented a subset of the language that can be effectively streamed [15]. A notable aspect of XSLT is that the W3C working group has looked to standardize a subset of the language that is appropriate for streaming applications.

6. UNCERTAIN STREAMS

Applications like location-based services (RFID) and text recognition (OCR) are driven by data that is low-level, imprecise, and sequential. To effectively exploit this low-level data, it must be transformed into higher-level data that is meaningful to a particular application. For example, in RFID applications, a sequence of raw sensor readings is transformed into a sequence of physical locations. In OCR, the low-level sequence of images on a page is transformed into a sequence of ASCII characters. Often, this transformation uses a probabilistic model like a Hidden Markov Model for RFID, Kalman Filter for tracking, Stochastic Transducer for Google's Ocropus tool for OCR, or approximates location data by uncertain ranges defined using continuous probability distributions over locations of moving objects. Besides the richness of data models, applications also need a variety of querying and monitoring facilities, such as continuous and probabilistic versions of spatial queries including nearest neighbour, range, and similarity queries, and queries specified by finite automata that can exploit the order of data items in the stream.

This workshop session featured three talks that covered complementary aspects of challenges in managing uncertain streams that are exemplary for most of the existing efforts in this research area.

Chris Ré overviewed work done in the Lahar [20] and Hazy research projects to effect transformations from low-level to high-level high quality uncertain streams modelled by Markov Sequences and subsequently to query such streams using transducers (i.e., automata with output) [13]. He presented several applications including a monitoring application based on uncertain RFID readings [20] and the GeoDeepDive application, which aims at unearthing data from the Geoscience literature by modelling OCR output using Stochastic Transducers and by integrating such models into relational database systems [14].

Reynold Cheng presented work on continuous nearest neighbour and range queries over imprecise location, where data is modelled by uncertain ranges defined by continuous probability distributions over locations of moving objects [7, 24]. In location-based services, saving communication bandwidth between servers and objects and mobile devices' battery is essential and Reynold showed how this can be effectively achieved by employing object filtering based on the probability that the object is close to a given query point.

Themis Palpanas surveyed techniques for modeling and processing data series with value uncertainty, an important model for temporal data, where each data point in the series is represented by an independent discrete or continuous random variable. He focused on the problem of answering similarity queries on uncertain data series, and described a novel technique for this problem [9]. In addition, he discussed the challenges of dealing with both value and existential uncertainty in processing streaming uncertain data.

7. STREAMING FRAMEWORKS AND SYSTEMS

A major goal of the workshop was to bring together, on the one hand, computer scientists working in particular stream-processing domains (XML, RDF, etc.) or particular streaming algorithms, with researchers studying broad stream-processing systems capable of expressing a wide range of applications. Nesime Tatbul's talk focused on relational stream processing engines. This included an overview of both language proposals, such as *STREAM CQL*, *StreamSQL*, and *MATCH-RECOGNIZE*, along continuous querying architectures, such as the DBMS-

based architectures of systems like *Truviso* and native streaming systems *StreamBase*. The ultimate goal would be to have an architecture that could express the features of each of the differing approaches to relational stream-processing, along with a clear set of systems definitions and embeddings of each engine into the "universal architecture". Tatbul's talk gave one step towards this goal, a versatile framework, *SECRET* [6], for describing the semantics of such systems, along with example descriptions of how some of the leading systems fit into the framework.

Yanif Ahmad talked about a new architecture being developed at Johns Hopkins for building next-generation streaming applications. Instead of beginning with "merely" data management infrastructure, the approach described by Ahmad begins with K3 [22], an event-driven language for general-purpose programming, building into the language both support for declarative data manipulation languages (e.g. for view definitions) and control structures for parallel and distributed programming.

8. DISTRIBUTED STREAMS

Big data analytics requires partitioning of large data streams into thousands of partitions according to specific set of keys so that different machines can continuously process different data partitions in parallel. This workshop session focused on analyzing requirements of and on solutions for distributed stream processing systems in the face of machine failure, pay-as-you-go models of computation, high-quality data partitioning, and low-overhead communication.

Peter Pietzuch discussed an approach to elastic and fault-tolerant stateful stream processing in the cloud, which was tested using the Linear Road Benchmark on the Amazon EC2 cloud platform [10]. The key aspects of this approach are on-demand scaling by acquiring additional virtual machines and parallelizing operators at runtime when the processing load increases, and fault-tolerance with fast recovery times yet low per-machine overheads.

Milan Vojnovic discussed the problem of range partitioning for big data analytics, where the goal is to produce approximately equal-sized partitions since the job latency is determined by the most loaded node [23]. The key challenge is to determine cost-effectively and accurately the partition boundaries in the absence of prior statistics about the key distribution over machines for a given input dataset. Cosmos, the cloud infrastructure for big data analytics used by Microsoft Online Services Division, uses a solution to this problem based on weighted sampling. Milan further presented a solution to the problem of continuous distributed counting [16], which had been mentioned earlier by Mila Hardt in her talk about Twitter.

Minos Garofalakis overviewed his recent work on approximate query answering with error guarantees in a distributed data streaming setting, where the focus is on communication efficiency, in addition to the standard space and time-efficiency requirements. In particular, Minos talked about sketching for distributed sliding windows [17], tracking complex aggregate queries [8], sketches based on the Geometric method, and sketch prediction models [11].

9. ACKNOWLEDGMENTS

We would like to thank the Engineering and Physical Sciences Research Council of the UK, who have sponsored INQUEST as part of the project *Enforcement of Constraints on XML Streams*, EPSRC EP/G004021/1.

Many of the staff at University of Oxford's computer science department were instrumental in making the workshop happen. In particular, we are very grateful to Polly Dunlop and Elizabeth Walsh for managing all of the arrangements for the meeting. We also thank Christoph Haase for being the webmaster for INQUEST.

10. REFERENCES

[1] Foteini Alvanaki, Sebastian Michel, Krithi Ramamritham, and Gerhard Weikum. EnBlogue: emergent topic detection in web 2.0 streams. In *SIGMOD*, 2011.

[2] Foteini Alvanaki, Sebastian Michel, Krithi Ramamritham, and Gerhard Weikum. See what's enblogue: real-time emergent topic identification in social media. In *EDBT*, 2012.

[3] Darko Anicic, Sebastian Rudolph, Paul Fodor, and Nenad Stojanovic. Stream reasoning and complex event processing in ETALIS. *Semantic Web*, 3(4), 2012.

[4] Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, and Michael Grossniklaus. C-SPARQL: a continuous query language for RDF data streams. *Int. J. Semantic Comp.*, 4(1), 2010.

[5] David Basin, Felix Klaedtke, and Samuel Müller. Policy monitoring in first-order temporal logic. In *CAV*, 2010.

[6] Irina Botan, Roozbeh Derakhshan, Nihal Dindar, Laura M. Haas, Renée J. Miller, and Nesime Tatbul. SECRET: a model for analysis of the execution semantics of stream processing systems. *PVLDB*, 3(1), 2010.

[7] Jinchuan Chen, Reynold Cheng, Mohamed F. Mokbel, and Chi-Yin Chow. Scalable processing of snapshot and continuous nearest-neighbor queries over one-dimensional uncertain data. *VLDB J.*, 18(5), 2009.

[8] Graham Cormode and Minos N. Garofalakis. Streaming in a connected world: querying and tracking distributed data streams. In *EDBT*, 2008.

[9] Michele Dallachiesa, Besmira Nushi, Katsiaryna Mirylenka, and Themis Palpanas. Uncertain time-series similarity: Return to the basics. *PVLDB*, 5(11), 2012.

[10] Raul Castro Fernandez, Matteo Migliavacca, Evangelia Kalyvianaki, and Peter Pietzuch. Integrating scale out and fault tolerance in stream processing using operator state management. In *SIGMOD*, 2013.

[11] Nikos Giatrakos, Antonios Deligiannakis, Minos N. Garofalakis, Izchak Sharfman, and Assaf Schuster. Prediction-based geometric monitoring over distributed data streams. In *SIGMOD*, 2012.

[12] Lukasz Golab, Howard J. Karloff, Flip Korn, Barna Saha, and Divesh Srivastava. Discovering conservation rules. In *ICDE*, 2012.

[13] Benny Kimelfeld and Christopher Ré. Transducing markov sequences. In *PODS*, 2010.

[14] Arun Kumar and Christopher Ré. Probabilistic management of OCR data using an RDBMS. *PVLDB*, 5(4), 2011.

[15] Pavel Labath. Xslt streamability analysis with recursive schemas. In *RCIS*, 2012.

[16] Zhenming Liu, Bozidar Radunovic, and Milan Vojnovic. Continuous distributed counting for non-monotonic streams. In *PODS*, 2012.

[17] Odysseas Papapetrou, Minos N. Garofalakis, and Antonios Deligiannakis. Sketch-based querying of distributed sliding-window data streams. *PVLDB*, 5(10), 2012.

[18] Danh Le Phuoc, Minh Dao-Tran, Josiane Xavier Parreira, and Manfred Hauswirth. A native and adaptive approach for unified processing of linked streams and linked data. In *ISWC*, 2011.

[19] Daniel Preotiuc-Pietro, Sina Samangooei, Trevor Cohn, Nicholas Gibbins, and Mahesan Niranjan. Trendminer: An architecture for real time analysis of social media text. In *ICWSM*, 2012.

[20] Christopher Ré, Julie Letchner, Magdalena Balazinska, and Dan Suciu. Event queries on correlated probabilistic streams. In *SIGMOD*, 2008.

[21] Sina Samangooei, Daniel Preotiuc-Pietro, Jing Li, Mahesan Niranjan, Nicholas Gibbins, and Trevor Cohn. Regression models of trends in streaming data. Technical report, University of Sheffield, 2012.

[22] P. C. Shyamshankar, Zachary Palmer, and Yanif Ahmad. K3: Language design for building multi-platform, domain-specific runtimes. In *XLDI*, 2012.

[23] Milan Vojnovic, Fei Xu, and Jingren Zhou. Sampling based range partition methods for big data analytics. Technical Report MSR-TR-2012-18, Microsoft Research, 2012.

[24] Yinuo Zhang and Reynold Cheng. Probabilistic filters: A stream protocol for continuous probabilistic queries. *Inf. Syst.*, 38(1), 2013.

The relational model is dead, SQL is dead, and I don't feel so good myself

Paolo Atzeni Christian S. Jensen Giorgio Orsi Sudha Ram
Letizia Tanca Riccardo Torlone

ABSTRACT

We report the opinions expressed by well-known database researchers on the future of the relational model and SQL during a panel at the International Workshop on Non-Conventional Data Access (NoCoDa 2012), held in Florence, Italy in October 2012 in conjunction with the 31st International Conference on Conceptual Modeling. The panelists include: Paolo Atzeni (Università Roma Tre, Italy), Umeshwar Dayal (HP Labs, USA), Christian S. Jensen (Aarhus University, Denmark), and Sudha Ram (University of Arizona, USA). Quotations from movies are used as a playful though effective way to convey the dramatic changes that database technology and research are currently undergoing.

1. INTRODUCTION

As more and more information becomes available to a growing multitude of people, the ways to manage and access data are rapidly evolving as they must take into consideration, on one front, the kind and volume of data available today and, on the other front, a new and larger population of prospective users. This need on two opposite fronts has originated a steadily growing set of proposals for non-conventional ways to manage and access data, which fundamentally rethink the concepts, techniques, and tools conceived and developed in the database field during the last forty years. Recently, these proposals have produced a new generation of data management systems, mostly non-relational, proposed as effective solutions to the needs of an increasing number of large-scale applications for which traditional database technology is unsatisfactory.

Today, it is common to include all the non-relational technologies for data management under the umbrella term of "NoSQL" databases. Still, it is appropriate to point out that SQL and relational DBMSs are not synonymous. The former is a language, while the latter is a mechanism for managing data using the relational model. The debate on SQL vs. NoSQL is as much a debate on SQL, the language, as on the relational model and its various implementations.

Relational database management systems have been around for more than thirty years. During this time, several revolutions (such as the Object Oriented database movement) have erupted, many of which threatened to doom SQL and relational databases. These revolutions eventually fizzled out, and none made even a small dent in the dominance of relational databases. The latest revolution appears to be from NoSQL databases that are touted to be non-relational, horizontally scalable, distributed and, for the most part, open source.

The big interest of academia and industry in the NoSQL movement gives birth, once more, to a number of challenging questions on the future of SQL and of the relational approach to the management of data. We discussed some of them during a lively panel at the NoCoDa Workshop, an event held in Florence, Italy in October 2012 organized by Giorgio Orsi (Oxford University), Letizia Tanca (Politecnico di Milano) and Riccardo Torlone (Università Roma Tre). We have used a provocative title (paraphrasing a quote often attributed to Woody Allen) and quotations from movies to elaborate on three main issues:

- the possible decline of the relational model and of SQL as a consequence of the rise of the non-relational technology,
- the need for logical data models and theoretical studies in the NoSQL world, and
- the possible consequences of sacrificing the ACID properties in favor of system performance and data availability.

In the following sections we discuss these issues in turn and close the paper with a final discussion. Since a consensus was reached on most of the issues addressed in the panel, we synthesize shared

opinions, rather than report contributions to the discussion by single individuals.

2. THE END OF AN ERA?

2.1 Relational databases

> "The ship will sink." "You're certain?"
> "Yes. In an hour or so, all of this will be at the bottom of the Atlantic."
> (Titanic. 1997)

According to Stonebraker et al., RDBMS are 25-year-old legacy code lines that should be retired in favor of a collection of from-scratch specialized engines [9]. Are we really attending the sinking of the relational ship?

One needs to distinguish between the relational model and its dominant query language, SQL, on the one hand and relational database management systems on the other.

The relational model and SQL were invented at a time when data management targeted primarily administrative applications. The goal was to support applications exemplified well by banking. The data is well structured: accounts, customers, loans, etc. And typical transactions include withdrawals and deposits that alter account balances. The relational model and SQL are well suited for managing this kind of data and supporting workloads made up from these kinds of transactions.

However, the data management landscape has evolved, and today's landscape of data management applications is much more diverse than it was when the relational model and SQL were born. Examples of this diversity abound: semi-structured data, unstructured data, continuous data, sensor data, streaming data, uncertain data, graph data, and complexly structured data. Similar diversities can be found in the workloads to be supported today.

Thus, while relational database systems were first proposed as a way to store and manage structured data, burgeoning NoSQL databases, such as CouchDB, MongoDB, Cassandra, and Hbase, have emerged as a way to store unstructured data and other complex objects such as documents, data streams, and graphs. With the rise of the real-time web, NoSQL databases were designed to deal with very large volumes of data.

Moreover, while relational database systems are usually scaled up (i.e., moved to larger and more powerful servers), NoSQL database systems are designed to scale out, i.e, the database is distributed across multiple hosts as load increases. This is more in line with real time web traffic as transaction rates and availability requirements increase and as data stores move into the cloud. The new breed of NoSQL systems are designed so they can easily scale up using low cost commodity processors to yield economic advantages.

Next, the data management applications have not just grown to concern more diverse kinds and uses of data. They have also become more complex. A single application may involve diverse kinds of data. This means that it is generally not possible for an application to use the single model and query language that is best for a single kind of data.

There are indeed two different issues here, related to the model level and to the implementation. In terms of implementation, it is clear (and it has been clear for more than a decade) that different applications have different requirements, especially when performance is a concern. This has led for example to separating OLTP and OLAP applications, even when the latter makes use of data produced by the former. Further, different engines with different capabilities have been developed for the two worlds, with specific support, the ones with more support for throughput of transactions and the others with support for very complex queries. With respect to models, the point is that most applications do need mainly simple operations over models that are somehow more complex than the relational one. NoSQL systems try to respond to these needs: implementations are new and specialized, operations are very simple, and diverse models (see the discussion on heterogeneity below) share the idea of being flexible (semistructured and with little or no schema).

2.2 SQL

> "Whoa, lady, I only speak two languages, English and bad English."
> (The Fifth Element. 1997)

A variety of data models and access methods are emerging and SQL is not suitable for any of them. Are we building the Babel Tower of query languages?

SQL has several advantages — it is a simple yet powerful declarative language for set-oriented operations. SQL captures the essential patterns of data manipulation, including intersections/joins, filters, and aggregations or reductions. Programmers who profess a dislike for SQL appear to have been deceived by its simplicity. The existence of languages such as SQLDF [4], which allows SQL queries on R data frames, add SQL functionality for analytics on Big Data. SQL's declarative expressions are

frequently more readable and compact than their R programmatic equivalents. Powerful extensions to SQL, based on window functions, provide a "split-apply" functionality otherwise known as map function. Combining these with SQL's GROUP BY operation, which is in reality a reduce function, essentially provides the equivalent of operations such as those in the Map Reduce framework.

However, in spite of the research and development, the relational model and SQL may not be the best foundation for managing every new kind of data and workload. The SQL-86 standard was a small and simple document. Then came SQL-89, SQL-92, SQL:1999, SQL:2003, SQL:2006, and SQL:2008. The current standard, SQL:2011, is very complex, and most data management professionals will find it challenging to understand. How many people have read and understood the entire SQL standard? Few claim that SQL is an elegant language characterized by orthogonality. Some call it an elephant on clay feet. With each addition, its body grows, and it becomes less stable. SQL standardization is largely the domain of database vendors, not academic researchers without commercial interests or users with user interests. Who is that good for?

Another aspect is that the SQL syntax requires the use of joins, considered ill-fit for, e.g., preferences and data structures for complex objects or completely unstructured data: many programmers would prefer to not do joins at all, keeping the data in a physical structure that fits the programming task as opposed to extracting it from a logical structure that is relational. Complex objects that contain items and lists do not always map directly to a single row in a single table, and writing SQL queries to grab the data spread out across many tables, when all you want is a record, is inconsistent with the belief that data should be persisted the way it is programmed.

On the other hand, the tumultuous developments we are observing have generated dozens of systems each with its own modeling features and its own APIs [2, 8], and this is definitely generating confusion. Indeed, the lack of a standard is a great concern for companies interested in adopting any of these systems [7]: applications and data are expensive to convert and competencies and expertise acquired on a specific system get wasted in case of migration. Efforts that support interoperability and translation are definitely needed [1]. Original approaches in this direction are needed, given the simplicity of operations and the almost total absence of schemas.

3. MODEL, THEORY AND DESIGN

3.1 Logical data models

> "Underneath, it's a hyper-alloy combat chassis, microprocessor-controlled. But outside, it's living human tissue: flesh, skin, hair, blood." (Terminator. 1984)

Aren't NoSQL database models too close to the physical data structures? What about physical data independence?

The ANSI SPARC architecture for database systems was defined in 1975 with the fundamental goal of setting a standard for data independence for DBMS vendor implementations. It appears that current NoSQL systems make no distinction between the logical and physical schema. Thus, the fundamental advantages of the ANSI SPARC architecture have been voided, which complicates the maintenance of these databases. Storing objects as they are programmed essentially negates the data independence requirement that then remains to be adequately addressed for NoSQL database systems. Strong typing of relations also allows definition of a variety of integrity constraints at the schema level, a very important consideration for transaction processing systems that support a variety of read, write, delete, and update transactions.

Relational database systems are criticized for the strong typing of relational schemas, which makes it difficult to alter the data model. Even minor changes to the data model of a relational database have to be carefully managed and may require downtime or reduced service levels. NoSQL databases have far more relaxed — or even nonexistent — data model restrictions. NoSQL Key Value stores and document databases allow applications to store virtually any structure it wants in a data element. Even the more rigidly defined BigTable-based NoSQL databases (Cassandra, HBase) typically allow new columns to be created with little effort. Actually, organizations should carefully evaluate the advantages and limitations of each type of systems (i.e. relational and NoSQL) for Big Data and then make an informed decision.

A common, high level interface could really be of use here. However it has to be simple, especially in terms of operations, as is the case for NoSQL systems. It is also worth mentioning that developers of the various systems follow "best practices" that support efficient execution of operations. An effort should be made to design a common interface by using the best practices of each system, with the goal of re-achieving physical independence.

3.2 Database theory

> "I've seen things you people wouldn't believe. [...] All those moments, will be lost in time, like tears in rain. Time to die." (Blade Runner. 1982)

Do we still need theoretical research in the new world? Has relational database theory become irrelevant?

The introduction of the relational model in 1970 marked a striking difference with respect to all the previous research on databases. The main reason for this lies in the strong mathematical foundations upon which this model is based, which provided the database research community with the possibility to approach the problems that were raised during the years by means of logical and mathematical tools, and to ensure the correctness and effectiveness of the proposed solutions by solid mathematical proofs.

This approach has caused the blooming of generations of splendid theoreticians who have set the foundations of the relational model, but have also contributed to adapting their experience to devise new methods and techniques for solving the problems derived from the advent of new challenges. Consider for instance the introduction of new paradigms for representing and querying semi-structured and unstructured data: since the nineties, invaluable theoretical research has laid the foundations for dealing with XML and the related query languages, with HTML Web data, with the Semantic Web, and with unstructured data like images and videos. It would be interesting to see what the work on semi-structured data and XML (modelling and languages) can contribute in the setting of NoSQL databases, since after all many of the problems rising from this new data model(s) have been discussed already within the semi-structured data research.

The lessons learned from developing the relational database theory have probably laid the methodological foundations for approaching most data-related problems, since, however unstructured and unkempt the datasets at hand, the understanding developed within the community will ever inform its research strategies.

3.3 Database design

> "They rent out rooms for old people, kill'em, bury'em in the yard, cash their social security checks." (No Country for Old Men. 2007)

How is database design affected by the recent paradigm shifts on logical data modeling? Is conceptual database design really too old for this country?

The methodological framework consisting of conceptual data modeling followed by the translation of the ER (or class-diagram) schema into a logical (relational) one can still be adopted: after all, these systems have to be accessed by applications. So, even if there is no schema in the data store, it is very likely that the data objects belong to classes, whose definitions appear in the programs, so some contribution could arise. At the same time, flexibility is a must, as objects could come from classes in an inheritance hierarchy, so polymorphism should be supported. The availability of a high-level representation of the data at hand, be it logical or conceptual, remains a fundamental tool for developers and users, since it makes understanding, managing, accessing, and integrating information sources much easier, independently of the technologies used.

4. ACID OR AVAILABLE?

> "Ask me a question I would normally lie to." (True Lies. 1994)

A relational database is a perfect world where data is always consistent (even if not true). Are the ACID properties really less relevant in modern database applications? Are we ready for a chaotic world where data is always available but only "eventually" consistent?

While preserving ACID properties may not be as important for databases that typically contain append only data, they are absolutely essential for most operational systems and online transaction processing systems, including retail, banking, and finance. ACID compliance may not be important to a search engine that may return different results to two users simultaneously, or to Amazon when returning sets of different reviews to two users. In these applications, speed and performance triumph the consistency of the results. However, in a banking application, two users of the same account need to see the same balance in their account. A utility company needs to display the same "payment due amount" to two or more users perusing an account. The idea of "eventual consistency" for such applications could lead to chaos in the business world. Is it by chance that just those applications that need full consistency are often those that better match the relational structure? Can we imagine a bank, a manufacturing or a commercial company which would rather use a complex-object data model to represent their data? This is probably why many

people mix up the structure of the relational model with the ACID properties, which in principle are completely independent aspects.

A consequence of the choices made in some systems about weak forms of consistency is that the burden is passed to applications developers, when they need to ensure more sophisticated transaction properties.

An observation that has been recently made about transaction management (and other implementation issues) is related to the fact that it can be easy to omit features, as this simplifies the development, but it might be difficult to reintroduce them later. Mohan [6] points out that there were experiences in the past with similar simplifications, and it was later very complex to obtain more general and powerful systems— some features needed to be rewritten from scratch.

5. FINAL COMMENTS

> "Look! It's moving. It's alive!!"
> (Frankenstein. 1931)

In spite of the shortcomings and inadequacies of the relational model and SQL, these technologies are, however, still going strong. Why? A key reason is that the systems that implement these are plentiful and have proven their worth. Perhaps the most important reason is that enormous investments are sitting in applications built on top of such systems. Companies around the globe rely on these applications and their underlying database management systems for their day-to-day business. Actually, relational DBMS provide the most understandable format for business application data, and at the same time guarantee the consistency properties that are needed in business. In addition, the skill sets of their current and prospective employees are targeted at these systems. It is not an easy decision to throw away relational and SQL technology and instead adopt new technology. Rather, it is much easier to extend the current applications and systems with no radical changes. Indeed, to the extent applications involve standard administrative data and "new" data, relational technology may even be best suited.

Thus, when is it reasonable for an organization to bet on a tool that is slightly incompatible with all the others, may be built by a community in open source model, does not grant consistency and concurrency control and is subject to change, neglect, and abandon at any point in time? The point is that there are killer applications – e.g. storing huge amounts of (read-only) social-network or sensor data in clusters of commodity hardware – that may make it worthwhile.

Therefore, we all believe that relational and NoSQL database systems will continue to coexist. In the era of large, decentralized, distributed environments where the amount of devices and data and their heterogeneity is getting out of control, billions of sensors and devices collect, communicate and create data, while the Web and the social networks are widening the number of data formats and providers. NoSQL databases are most often appropriate for such applications, which either do not require ACID properties or need to deal with objects which are clumsily represented in relational terms.

As a conclusion, NoSQL data storage appears to be additional equipment that business enterprises may choose to complete their assortment of storage services.

With all these questions ahead the contribution the database community can give is huge. Let us take a full breath and start anew!

6. REFERENCES

[1] P. Atzeni, F. Bugiotti, and L. Rossi. Uniform access to non-relational database systems: The SOS platform. In *CAiSE 2012*, Springer, pages 160–174, 2012.

[2] R. Cattell. Scalable SQL and NoSQL data stores. *SIGMOD Record*, 39(4):12–27, 2010.

[3] M. Driscoll. SQL is Dead. Long Live SQL! `http://www.dataspora.com/2009/11/sql-is-dead-long-live-sql/`, 2009.

[4] G. Grothendieck. SQLDF: SQL select on R data frames. `http://code.google.com/p/sqldf/`, 2012.

[5] G. Harrison. 10 things you should know about NoSQL databases. `http://www.techrepublic.com/blog/10things/10-things-you-should-know-about-nosql-databases/1772`, 2010.

[6] C. Mohan. History repeats itself: sensible and NonsenSQL aspects of the NoSQL hoopla. In *EDBT 2013*, ACM, pag. 11–16, 2013.

[7] M. Stonebraker. Stonebraker on NoSQL and enterprises. *Commun. ACM*, 54:10–11, 2011.

[8] M. Stonebraker and R. Cattell. 10 rules for scalable performance in 'simple operation' datastores. *Commun. ACM*, 54(6):72–80, 2011.

[9] M. Stonebraker, S. Madden, D. J. Abadi, S. Harizopoulos, N. Hachem, and P. Helland. The end of an architectural era: (it's time for a complete rewrite). In *VLDB 2007*, VLDB Endowment, pag. 1150-1160, 2007.