Department of
Information Engineering
and Computer Science DISI

UNIVERSITY
OF TRENTO - Italy

DISI - Via Sommarive, 14 - 38123 POVO, Trento - Italy
http://disi.unitn.it

ResEval: An Open and Resource-oriented Research Impact Evaluation tool

Muhammad Imran, Maurizio Marchese, Azzurra Ragone, Aliaksandr Birukou, Fabio Casati, Juan Jose Jara Laconich

February 2010

Technical Report # DISI-10-016

Abstract

In this paper we present the concepts and the architecture of an open and resource-oriented tool for evaluating research impact of individual researchers and groups. The advantages of the tool are that it relies on several sources for computing metrics, freely available on the Internet, and provide an easy way for defining and computing custom metrics for individuals and research groups.

1 INTRODUCTION

Today there is a great pressure in academia and research centers to **measure** the productivity of **researchers** or researcher groups, in order to decide which people to hire or promote, or how to divide funds among groups e.g. in a department or among universities [11]. Scientific research is heavily funded by governments and institutions around the world, that want to be able to have some *reliable, personalized* and *flexible* metrics to monitor both the productivity of their public money and the quality/impact of research, in order to establish the policies for future investments.

Another important dimension in the research impact evaluation domain lies in the exponential growth of the amount of available scientific/scholarly digital content. This leads to the need to have reliable, fast and - as much as possible - automated tools to support the query and successive ranking of "interesting" scientific contributions and researchers. Also in this case, the availability of personalized and flexible metrics based on the potentially available digital sources could support scientists in their search for high impact scientific artifacts (not only papers, but also shared experimental data and procedures, influential blog entries and interesting discussions). Moreover, one could search for the most influential researchers in the DB community or do a search for the "best" paper in some topic across different communities.

In this paper we present some preliminary concepts and tools, exploiting new opportunities and lessons learned from Web 2.0 and service-oriented paradigm, for an efficient (for people), effective (for science), and sustainable (for the community) way of evaluating scientific research impact. Specifically we present and discuss a model, languages and architecture for a tool capable of evaluating research contributions and people by using personalized and pluggable metrics on data gathered automatically from a variety of scientific information sources.

Our aim is to have a web application that can support in the semi-supervised evaluation of the research impact of a certain contribution, researcher and/or groups of researchers (e.g., departments, universities, research center, countries). The main contributions of the proposed approach are:

- A common platform to access various kinds of scientific resources available on the web, supporting a simple framework to develop value added services for scientists and evaluation centers. Such a platform provides programmatic and universal access to scientific resources, hiding the tedious problem of accessing heterogeneous services which very often are not even available for programmatic access but are only designed for Web browser access (e.g., Google Scholar).

- A tool that supports the definition and implementation of tailored and personalized metrics that can be easily integrated in a service-oriented platform.

- The capability to define in specific Research Evaluation Languages the personalized metrics, the query on web sources and the group definition and selection.

- The capability to evaluate not only individual scientists or contributions, but also research groups, departments and institutions.

The rest of the paper is organized as follow: in the next section we briefly revise some related work, and then we provide a description of the conceptual model, languages, and architecture. In Section 3 we present the prototype of the proposed evaluation tool and preliminary results. Some conclusion and discussion of future work closes the paper.

1.1 RELATED WORK

Today there is no unique and widely accepted way to assess the productivity of a researcher. Some simply use the number of publications or grants received. Other use somewhat widely accepted metrics based on citations.

Bibliometrics indexes have become a standard and popular way to assess research impact in the last few years. All significant indicators heavily rely on publication and citation statistics and other, more sophisticated bibliometric techniques. In particular, the concept of citation [7] became a widely used measure of the impact for scientific publications although problems with citation analysis as a reliable method of measurement and evaluation have been acknowledged throughout the literature [5]. Indeed, not always a paper is cited because of its merits, but also for some other reasons, as flaws, drawbacks or mistakes.

A number of other indices have been proposed to balance the shortcomings of citations count and to “tune” it so that it could reflect the real impact of a research work in a more reliable way. The most commonly used are h-index [9], h-b index [3], g-index [6], AR-index [10], contemporary h-index [12] and individual h-index [4].

For a number of years researchers had only one source for looking at this type of information: the commercial ISI Thomson Scientific database (currently Web of Science[1]). As authors in [11] have shown, this source cannot be used for evaluating computer science. In the following years, many other competitors emerged like Citeseer[2] , Scopus[3] , and Google Scholar[4] , with the purpose of giving users a simple way to broadly search the scholarly literature.

Based on the existing sources, new tools are beginning to be available to support people in the research impact analysis. A useful tool is Publish or Perish [8], a desktop based software program that uses Google Scholar to retrieves the citation data, and then analyzes it to generate the citations based metrics. Current weakness of this tool are: (i) the reliance on only one information source (Google Scholar); (ii) the need for manual cleaning of the obtained data (for example for author disambiguation and self-citations among others) and (iii) the lack of Application Programming Interface (API) over which other applications or web services could use the offered functionalities. A different approach is provided by Scholarometer[5]: a social tool which is used in citation analysis and also for evaluation of the impact of an author's publications. It is a browser free add-on for Firefox that provides a smart interface for Google Scholar and requires users to tag their queries with one or more discipline names. This generates annotations that go into a centralized database, which collects statistics about the various disciplines, such as average number of citations per paper, average number of papers per

[1]http://scientific.thomson.com/products/wos/

[2]http://citeseer.ist.psu.edu/

[3]http://www.scopus.com/home.url

[4]http://scholar.google.com/

[5]http://scholarometer.indiana.edu/

authors, etc. The impact measures are then dynamically recalculated based on the user's manipulations. Scholarometer provides a server where information about the queries performed and their results are stored. However, it does not offer an API to retrieve or use this information.

2 PROPOSED SOLUTION

Even though there are applications for assessing research impact, all the currently available solutions/tools lack in our view some key features, namely: (1) completeness of data, (2) flexible and personalized metrics (3) languages to support the user in defining sources, queries and metrics and (4) data processing options. Data completeness is indeed a main issue in the process of evaluating research people: e.g. some sources do not (completely) cover some disciplines, for instance Web of Science is not good for Computer Science, while it is very important to compute citations received by all the documents published by a given author. We tackle this issue by leveraging on an open, resource-oriented Resource Space Management System (RSMS) that is able to provide homogeneous programmatic access to heterogeneous resources and web services, regardless of how they are implemented, as long as they are web accessible.

The possibilities to define flexible metrics is essential in order to have a personalized access to the information, e.g. one might want to exclude self-citation from the h-index value of a researcher or see how this index could change excluding citations coming from the top co-authors. For research group evaluation one might want to specify the community in order to assess in the proper context the group impact. We take this challenge by using a service-oriented architecture that allows users to use existing metrics or define their own metrics.

To assist the user we are developing specific Research Evaluation Languages in order to support the user in both the query and the definition of the personalized metrics. Moreover, we provide further data processing options by supporting the user in the creation of communities to be evaluated and analyzed. In the following we briefly describe the three layers present in the proposed solution: Concepts, Languages and Architecture.

2.1 Concepts

The main concepts of our proposed approach are:

- The use of the web as a source of information. We take advantage of the already available information and use a homogeneous programmatic access to scientific resources. We base our approach on current research on Resource Space Management System.
- The use of service oriented architecture to increase the extensibility of our web application that allows the development of widgets and vertical applications tailored to the specific user's needs.
- The implementation of all modules as services to improve interoperability between them.
- The definition of specific Research Evaluation Languages in order to support the user in both the query and the definition of the personalized metrics.

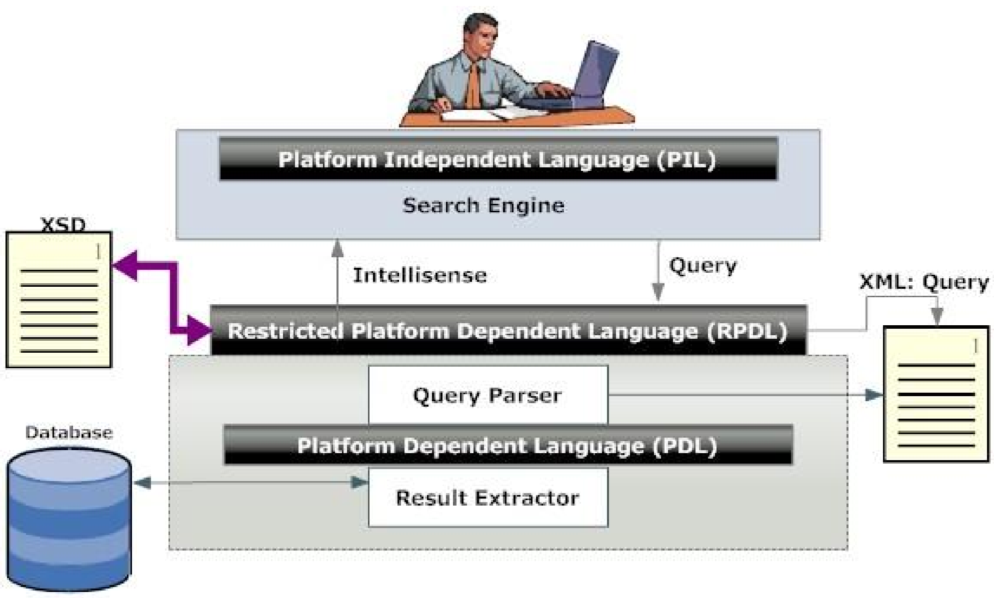

Figure 1: Structure of a REL

2.2 Research Evaluation Languages

The main goal of Research Evaluation Languages (RELs) is to provide a simple yet powerful interface to nontechnical user, so that they can define metrics and communities without any DBMS or algorithmic related knowledge. One of the important aspects in this context is application of natural language interface (NLI). The RE languages themselves are defined using XML Schema (XSD).

We divided REL structure in three layers (figure 1). A Platform Independent Language (PIL) that allows users to use a restricted natural language to write queries to define metrics and groups. There have been several approaches working towards use of Natural Languages [2], but the best applications so far use some restrictions in the expression power of the user [1]. As an example, for the case of queries, users are given support to write simple or complex queries. This query is in the form of a Restricted Platform Dependent Language (RPDL). The query is then transformed into XML, and then it is converted into a Platform Dependent Language (PDL) which is in our case SQL.

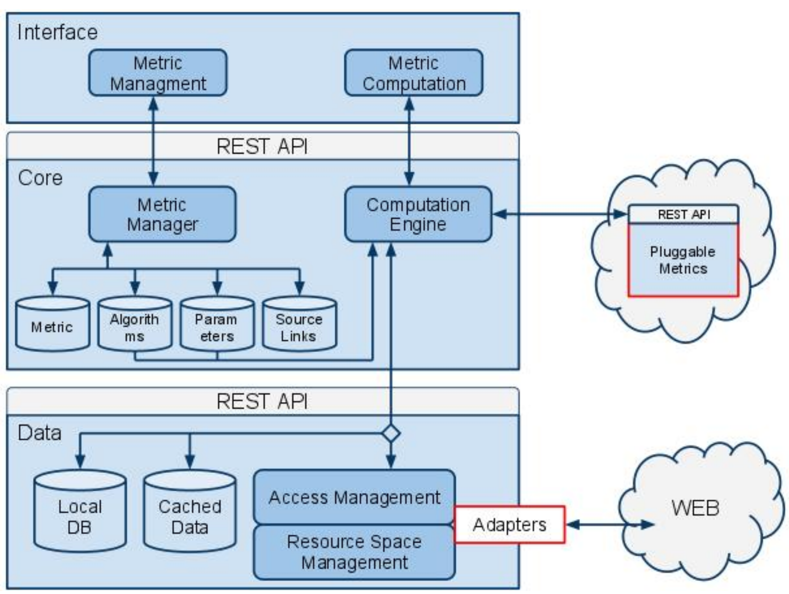

Figure 2: Main Architecture

2.3 Architecture

The architecture of the ResEval tool is defined in three well-differentiated layers, as seen in Figure 2.

Interface layer: this is the layer that provides the main functionalities of the tool through several user interfaces. One interface offers all the functionalities needed to create new metrics or manage existing ones. The other let us to compute existing metrics defined using the available options and source of information (e.g. Scholar Google). The choice of developing multiple interfaces is also motivated by the need to have, eventually, the possibility to embed the application in a more complex platform.

Core layer: in this layer the tool stores and manages all the definitions and logic of the metrics. Metric manager is the core component of the tool, the one that gets the metric definition, retrieves all the data needed to compute a metric from the specified source, and finally computes it. The computation engine component computes the metrics using available options according to metric definition and information sources. Once a metric is completely defined, this component will provide the user with the interface necessary to compute it. The modules that can be identified in a metric architecture, and that can be seen in Figure 3, are the following:

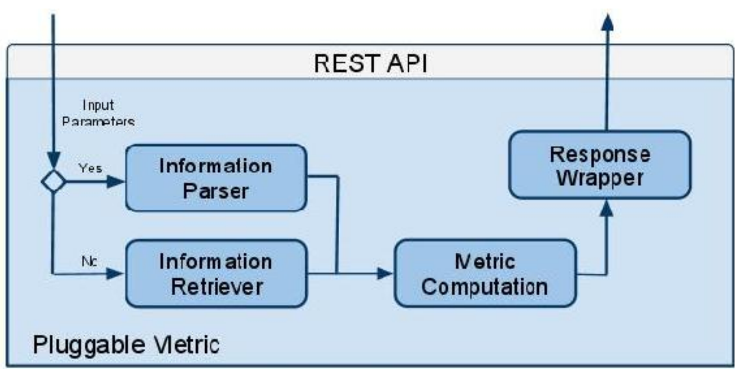

Figure 3: Pluggable metric architecture

- *Information Parser:* is the module that parses the information received from the application and prepares it for the metric computation.

- *Information Retriever:* is the module that gets the information from the metric's own source and prepares it for the metric computation.

- *Metric Computation:* is the module that allocates and executes the algorithm that computes the metric.
- *Response Wrapper:* is the module that prepares the response message that will be returned to the application with the result of the metric.

The parameters that need to be set for computing a metric are: Object (the name of the scientific object that is evaluated), Object Type (the type of the object that is evaluated, currently author and contribution types are supported), Metric (the name of the metric that is computed for the selected object), Algorithm (the name of the algorithm that is used to compute the selected metric), Source (the name of the source from where the information needed to compute the metric is retrieved), Source filters (the set of filtering options that are used to narrow the number of results from the selected source), Reload (flag that indicates the tool to get recent information from the source instead of cached one).

Data layer: this layer is used to get the data needed in order to compute a specific metric. There are different sources where the tool can get the data: internal sources, such as a local database, or external sources found in the web. For these external sources an adapter must be implemented that is able to retrieve the information for each source. Furthermore, the adapter must be registered in the resource space management and access management modules in order to be available for use. The architecture includes the cache mechanism to store the information retrieved from underlying sources. The caching system is clearly motivated by performance reasons, since we are dealing with a web application that often queries external servers and, therefore, loses much time due to network's overhead.

3 ResEval Restful API & PROTOTYPES

At present we have implemented two prototypes based on ResEval RestFul API. The first prototype provides a simple web application to query for the evaluation of research contribution and individual researcher using citation based metrics and can be used and tested at `http://project.liquidpub.org/reseval/`. At present, we have implemented (as plug-in) the following standards metrics for individual researchers: overall citations count; number of publications; average number of citations per publication; h-index;

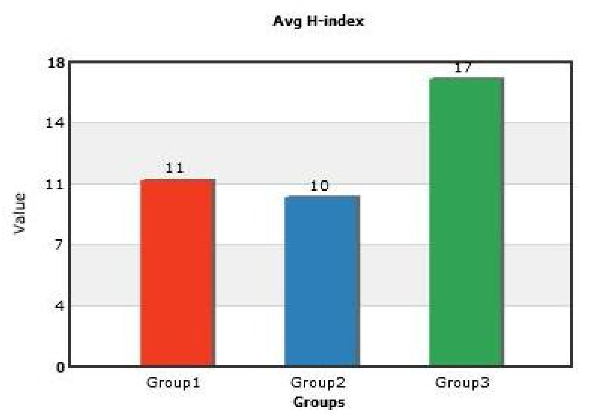

Figure 4: Average h-indexes comparison

g-index. In addition we also provide pluggable metric for some novel indicators, namely:

Signal to noise ratio: it's the ratio between the number of publications that contributes to the computation of h-index and g-index, i.e. those that have received at least one citation and the total number of publications. The aim of this metric is to somehow estimate the quality of the work of a researcher compared to the number of publications he wrote.

Top citers: it's a list of those that have cited the specified author the most. Users can click on the checkboxes to exclude one or more of them, in order to see how the indexes change without their citations. The idea is to see whether the citations come only from people "known" by the specified author (i.e. co-authors, himself) or, conversely, the author is cited by people from different departments, organizations, universities.

Top co-authors: it's the list of those that more often have co-authored a publication with the specified author. Users can click on the check boxes to exclude one or more of them. This aims to be an indicator of the independence of the author. ***h-index and g-index without self citations:*** self citations are subtracted to the total number of citations, then the two indexes are recomputed. This is used to see how much the self-citations weigh on the assessment of the specified author we can compute. The second prototype is a group comparison web application, available at `http://project.liquidpub.org/groupcomparison/`. It allows users to create groups of researchers and to do comparison across and within groups. A

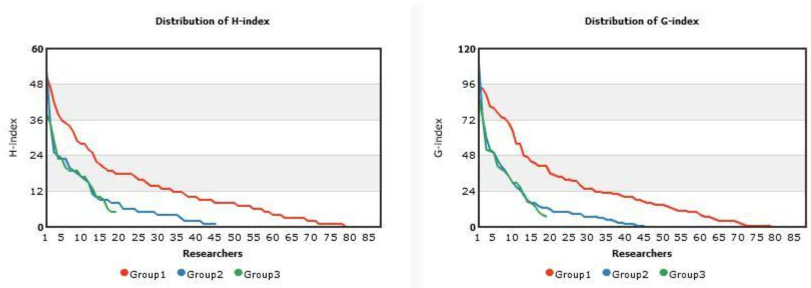

Figure 5: Example of group comparison: distribution of h- indexes and g-indexes for three groups.

group can be formed in various ways, e.g. finding all co-authors of a certain author, or considering all the full professors in a department. Once groups are created they can be compared considering e.g. average h-index, average g-index, number of publications, etc. An example of these analyses is shown in Figure 4 where we compare average h-index computed with the ResEval Restful API among three research groups.

In Figure 5 , we compare instead the distribution of two indexes (h-index and g-index) in two research groups. Specifically, we plot the value of the indices computed with the ResEval Restful API for every person in the group (ordered from the highest index to the lowest. Indeed, it is possible to compare two or more groups using the aggregate bibliometric indexes, as well as to compare scientists within the same group. Moreover, doing comparisons within a particular group it is possible to identify the most prominent scientists within the group.

4 CONCLUSION

We presented the concept and the architecture of an open and resource-oriented research impact evaluation tool. Even though first prototypes have been developed and are accessible in the web, the work is still at a preliminary stage. Ongoing work includes a full development of the languages described in Section 2 as well as leveraging on the Resource Space Management System approach in order to increase the number of web sources.

References

[1] R. Altwarg. Controlled languages, an introduction. centre for language technology website on controlled natural languages. 2006.

[2] I. Androutsopoulos, G. Ritchie, and P. Thanisch. *Natural language interfaces to databases-an introduction*. Citeseer, 1994.

[3] M. Banks. An extension of the Hirsch index: Indexing scientific topics and compounds. *Scientometrics*, 69(1):161–168, 2006.

[4] P. Batista, M. Campiteli, O. Kinouchi, and A. Martinez. An index to quantify an individual's scientific research valid across disciplines. *Arxiv preprint physics/0509048*, 2005.

[5] A. Chapman. Assessing research: citation count shortcomings. *The Psychologist*, 2:336–44, 1989.

[6] L. Egghe. An improvement of the H-index: the G-index. *ISSI Newsletter*, 2(1):8–9, 2006.

[7] E. Garfield and A. Welljams-Dorof. Of Nobel class: a citation perspective on high impact research authors. *Theoretical Medicine and Bioethics*, 13(2):117–135, 1992.

[8] A. Harzing. Publish or perish. *Retrieved December*, 7:2007, 2007.

[9] J. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences of the United States of America*, 102(46):16569, 2005.

[10] B. Jin, L. Liang, R. Rousseau, and L. Egghe. The R-and AR-indices: Complementing the h-index. *Chinese Science Bulletin*, 52(6):855–863, 2007.

[11] A. I. Report. Research evaluation for computer science. 2008.

[12] A. Sidiropoulos, D. Katsaros, and Y. Manolopoulos. Generalized Hirsch h-index for disclosing latent facts in citation networks. *Scientometrics*, 72(2):253–280, 2007.