A Traceability-based Method to Support Conceptual Model Evolution

Marcela Ruiz

PROS Research Centre, Universitat Politècnica de València, Spain
lruiz@pros.upv.es

Abstract. Renewing software systems is one of the most cost-effective ways to protect software investment, which saves time, money and ensures uninterrupted access to technical support and product upgrades. There are several motivations to promote investment and scientific effort for specifying systems by means of conceptual models and supporting its evolution. As an example, the software engineering community is addressing solutions for supporting model traceability, continuous improvement of business process, organisational reengineering, information system maintenance, etc. Model-driven techniques have been developed in order to analyse systems raising the abstraction level of its specification. However, a support for conceptual model evolution by means of model-driven techniques is still needed. This thesis proposes a traceability-based method that involves model-driven capabilities for designing and providing guidelines, techniques, and tools to support conceptual model evolution. The main idea is to support information system analysts in the tasks related to: justify **why** the conceptual models have evolved, report and specify **what** elements have evolved, and guide **how** to carry out evolution in certain predefined organisational contexts. We plan to apply our method to guide the evolution of an E-shopping software. This way, we also provide mechanism to facilitate industrial adoption.

Keywords: conceptual model evolution, reengineering frameworks, traceability-based support, business process modelling, intentional modelling, pattern definition, delta analysis

1 Introduction

Software maintenance and information system evolution are activities that receive significant dedication by industry. This is one of the reasons that motivate the information systems engineering community to investigate in this area. Organisations are aware on the need to apply mechanisms and strategies in order to encompass processes and products in changing environments. For instance, in organisational context, companies need to rethink business processes, infrastructures, technologies, resources, etc. according to new demands from their environment or changes in their organisational objectives. Business processes should also be transformed to support the new processes and tasks that result from the involvement of new objectives or

goals in the organisation. Then, constant organisational change and its influence in processes and products must be considered as a fundamental rule of competitive strategy for continuous improvement [1]. For software systems, the high pressure of a very short time-to-market often forces developers to implement the code of the application directly, without using a disciplined development process, which may have disastrous effects on the quality and documentation of the delivered software application [2]. These practices have been the motivation for opening new research lines in order to support post-delivery life-cycle activities. Besides, with regard to the keynote of the ERCIM News 88 magazine[1], some of external drivers for changing software are innovation, cost reduction and regulation; factors that need to be supported by techniques, tools and methods.

The main goal of my PhD thesis is *to design a traceability-based method that involves model-driven capabilities in order to support conceptual model evolution*. The main idea is to provide a model-driven method that can be used by information system analysts in order to provide them with reports and evidences to help decision making in information system evolution contexts. This paper summarizes the author's PhD work and project, working for two years and a half, under the supervision of Dr. Sergio España Cubillo in the PROS Research Centre of the Universitat Politècnica de València.

2 Problem Description and Research Methodology

Traditionally in software system development, the evolution process and information system maintenance have been faced by means of the reengineering process, change specification, evolution metrics, goal-driven requirements engineering and model management. For these reason, we explore current solutions in these fields in order to find related research that confronts conceptual model evolution.

The reengineering process is commonly defined and widely used by the scientific community by means of the metaphor of the "horseshoe" model, which purpose is to present the reengineering process in a figure (the horseshoe is basically a left-hand side, a right-hand side and a bridge between the sides). In general terms, the left-hand side of the horseshoe model consists of an extraction from an existing system to get the system specification, the right-hand side consist of conventional software development activities, and the bridge between the sides consists of a set of transformations from the old system to the new one [4]. Both, the left-hand side and right-hand side represent different levels of abstraction of the system. Nowadays, the Object Management Group (OMG) is working on promote an industrial consensus on modernisa-

---

[1] The ERCIM News 88 special theme was "Evolving Software" 3. Visser, J., *Change is the constant*, in *ERCIM news - Special theme: Evolving Software*. 2012: Sophia Antipolis Cedex, France. p. 3.. The magazine put together a set of papers to give an overview of both traditional and emerging software engineering techniques, tools and approaches used by software evolution experts.

tion of existing application by means of the initiative named Architecture-Driven Modernisation (ADM) [5]. This initiative is based on the MDD paradigm to automate the horseshoe model. However, full support for the evolution process (the bridge between the sides) is still missing. The authors of [6] aimed to automate the horseshoe model, although it is not severely applied.

Goal-driven requirements engineering approaches faced goal modelling from different perspectives of use. Some of those uses are: understanding the current organisational situations and need for change, decision making, relating business goals to functional and non-functional system components and validation of compliance between system specification and stakeholders' goals [7]. Co-evolution approaches has been proposed in order to understand reciprocal evolution of system components [8]. Nevertheless goal specification related with change models and specification of evolution grains is still an open research field.

System change and stability analysis in order to derive or facilitate system evolution is confronted by [9]. A method to support the elicitation of evolution requirements and a generic syntax to specify them is explored in [10]. Also, metrics for classifying and measuring software evolution are analysed by [11]. Even though, specification of evolution in with formal conceptual models and measurement techniques to provide meaningful to kick start analysis is still needed.

Model management confront problems in many databases application domains (e.g. data warehousing, semantic query processing, meta-data management, meta-data integration, schema evolution etc.); research projects in this area are aiming at providing high-level abstractions artefacts in order to offer a generic solution [12-13]. Bernstein [14] presents a full description of all of the model management operators. Moreover, no complete frameworks to support enterprise information system evolution have been proposed yet.

The problems detected establish the motivations in which this PhD thesis is founded.

2.1 Research Questions Objectives and Means

We follow design science to classify our research questions in knowledge problems (KP) and practical problems (PP) [15]. This way, we are looking for highlighting our research results by means of producing useful artefacts. This thesis is focused on conceptual model evolution. To achieve the main goal, we conceive the following research questions:

- **RQ1** (KP). What elements are common in conceptual model evolution? The answer to this question should clarify terminology, stakeholders, and helps to establish a conceptual framework to facilitate reasoning about conceptual model evolution.

- **RQ2** (KP). Which are the current conceptual model evolution methods? The answer to this question should establish the state of the art about current conceptual model evolution support.
    - **RQ2.1** (KP). Which of these methods are model-driven oriented?
- **RQ3** (PP). How can be supported a conceptual model evolution method? The answer to this question refers to the main **goal** of this thesis.
    - **RQ3.1** (PP). What guidelines are needed in order to evolve conceptual models?
    - **RQ3.2** (PP). What techniques are needed in order to facilitate the use of the method?
    - **RQ3.3** (PP). What tools are needed in order to support the use of guidelines and techniques?
- **RQ4** (PP). How can possible scenarios be integrated in the conceptual model evolution method? The answer to this question refers the modules to support business process evolution, goal-driven evolution, and reengineering.
- **RQ5** (KP). How can the model-driven method to support conceptual model evolution be validated? The answer to this question should establish a validation framework to measure feasibility, trade-off and sensitivity.

Means

To achieve the main goal and solve the research questions, three main means are conceived: a) Expert views. My directors are experts to guide my decisions to provide solutions of the addressed problem. b) Technological support. We are expert in model-driven tools as Eclipse. This way, we have capabilities to provide tool support for the method. c) Collaboration with other research groups. Collaboration increases our perspectives to provide solutions. d) Action research. Our proposal is motivated by the needs of real information system analysts.

3 Research Methodology

This PhD project follows the design science framework to design a new artefact: a model-driven method to support conceptual model evolution. The research methodology is explained by means of regulative cycles that were conceived in order to answer the research questions. Fig. 1 presents the research methodology.

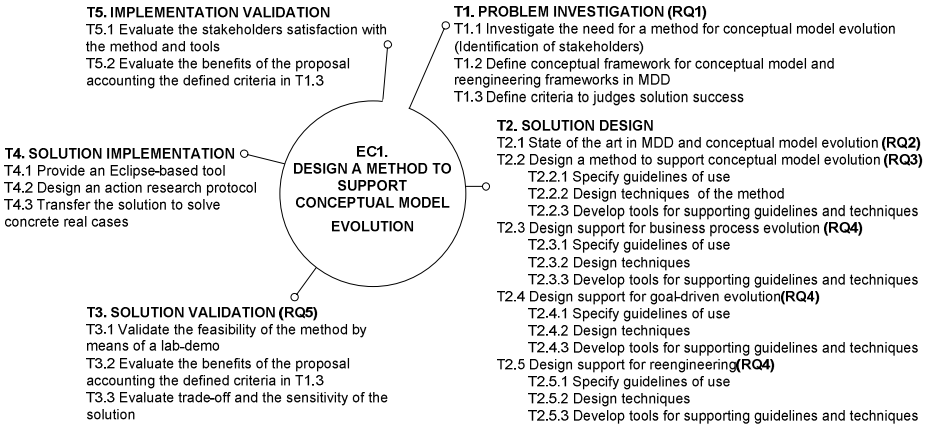

Fig. 1. Overview of the research methodology

Since our proposal focus on the development of a new artefact, the main cycle of the research methodology is an engineering cycle (EC1. Design a model driven method to support data system evolution). Concretely, this cycle is formed by 5 main tasks: T1) problem investigation; T2) solution design; T3) solution validation; T4) solution implementation; and T5) implementation validation.

An information system needs evolve. Since the information system is specified by means of models, we investigate current research to support conceptual model evolution. We identify the stakeholders or possible users of the method. To define the problem and define the method, we provide a conceptual framework to avoid terminology incoherence. In addition, we establish the criteria to judge the solution success when we finish the engineering cycle. These activities are related to T1.

In T2 we explore available solutions by reviewing state of the art. We design a new solution; i.e. our method. To do that, we design the guidelines of use; we provide techniques to facilitate the use of the method; and we develop tools (prototypes built in the laboratory) to support guidelines and techniques. Also, we design the support for the modules of business process evolution, goal-driven evolution and reengineering frameworks.

The method is validated in T3. We demonstrate the feasibility by means of lab-demo. We establish a comparative with the results of the lab-demo with the criteria defined in T1.3. Also, we evaluate trade-off and sensitivity of the solution.

In T4 we implement the method using Eclipse based tools, design an action research protocol to transfer the solution to be used in practice. Finally, in T5 we assess the operability of the tool, stakeholder's satisfaction and criteria of success by means the results of the action research protocol carried out in T4.

4 Proposal

We face the design of the method by two main motivations: 1) Market pull or demand pull and 2) Technology push [16]. The first one refers our motivation to evolve the E-Shopping software (a real case and we have into account the user needs). We call it market-driven solution. The second one refers our motivation to provide an invention without proper consideration of whether or not it satisfies a set of specific user needs. We call it technology push-driven solution.

To design the method, we have been inspired by the metaphor of a "horseshoe" of Kazman et. al. [4]. Carrying the horseshoe metaphor to the MDD field, an interesting evolution method can be provided for different scenarios. As a result, models are the main artefact and the analysis of them is in a high level of abstraction. The traceability-based support plays the main role in the method; it provides two types of traces: Vertical traces to relate elements that specify different characteristics of information systems (e.g., processes, goals, etc.); and horizontal traces are accounted to relate evolution of elements.

To use the method, the analyst should carry out the four tasks presented in the Fig. 2:

1. Define evolution question, in this task the analyst decides what characteristic of evolution process want to know. The analyst follows a set of guidelines in order to know if s/he wants to obtain information about justifying **why** the conceptual models have evolved, reporting or specifying **what** have evolved, or analysing **how** to evolve conceptual models according to a set of predefined solutions for certain contexts.

2. Specify As-Is and To-Be models, in this task the analyst specify the current and desired system to be analysed applying the evolution modules.

3. Apply evolution modules, in this task the analyst applies the module that corresponds with s/he evolution question.

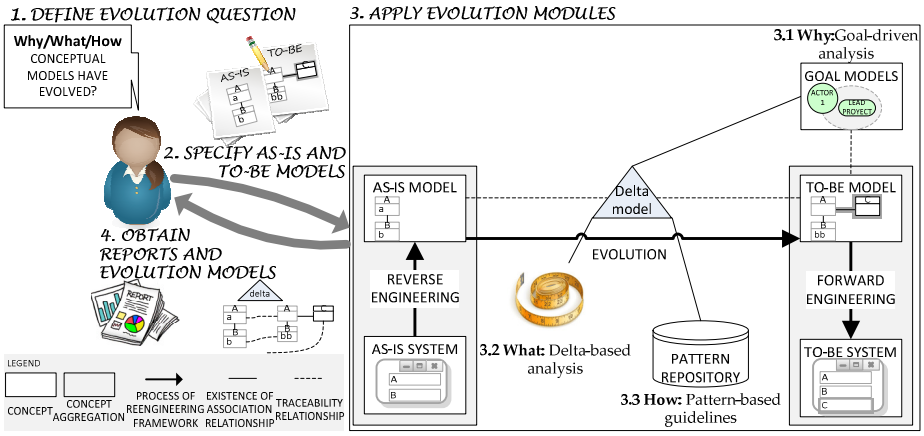

Fig. 2. Overview of the traceability-based Method to support conceptual model evolution

For the **why** question (*3.1 Why: Goal-driven analysis*), a goal driven model evolution support is provided. The vertical traceability is established between two information system specification languages. As a proof of concept, we have aligned the *i** framework with the Communication Analysis modelling techniques. Goal models are connected with delta models that specify changes in the information system.

For the **what** question (*3.2 What: Delta-based analysis*), a set of metrics are provided in order to report meaningful information about the evolution processes, elements involved and the conceptual impact of changes.

For the **how** question (*3.3 How: Pattern-based guidelines*), a set of patterns to evolve business process models have been established. The patterns are connected with delta models to register what changes implies the application of patterns.

4. Obtain reports and evolution models, in this task the analyst obtain the results of modules application. Based on the results the analyst can provide meaningful information about conceptual model evolution processes and make decisions based on evidences.

The method is in continuous improvement and re-adjusts. The modules have been designed; the implementation has being developed in Eclipse-based tools.

5 Progress of the Thesis

In 2012, organisational reengineering frameworks have been studied, focusing on RQ1 and RQ2. Furthermore, the alignment between the process and the goal perspectives were explored. As a proof of concept, we have aligned the *i** framework with the Communication Analysis modelling techniques. This proof of concept refers the RQ4. Also, we implemented the alignment of this modelling languages in an Eclipse-based tool (this implementation refers RQ3.). And we analysed the benefits and the limitations of aligning process and goal perspectives. We started a first version of the definition of the artefacts to support model evolution (Traceability support).

In 2013, the modules of the method were designed and reported. We carried out an experimental task with master students to analyse vertical traceability between conceptual models.

In 2014-2015 we plan to establish the method guidelines and delta analysis technique formalisation. In addition, we are looking for implementing pattern definition metamodel and evolution metamodel in an Eclipse plug-in (RQ3).We plan to validate the method and the prototype by means of laboratory demos. The idea is to estimate scalability, trade-off and sensitivity of our method. This validation refers RQ5.

We plan to finalize the implementation and the implementation validation of the method in 2015.

Acknowledgments
I acknowledge to my supervisor Sergio España for his invaluable support and advices to drive my thesis and encourage my research career.

This PhD project has been supported by the Spanish Generalitat Valenciana ORCA (PROMETEO/2009/015); the FPI grant of the Universitat Politècnica de València (3146); the European Commission FP7 Project CaaS (611351); and the ERDF structural funds.

References

1. Sage, A.P., *Systems Engineering and Systems Management for Reengineering.* Systems Software, 1995(30): p. 3-25.
2. Di Lucca, G.A., A.R. Fasolino, and P. Tramontana, *Reverse engineering Web applications: the WARE approach.* Journal of software maintenance and evolution: Research and practice, 2004. **16**(1-2): p. 71-101.
3. Visser, J., *Change is the constant*, in *ERCIM news - Special theme: Evolving Software*. 2012: Sophia Antipolis Cedex, France. p. 3.
4. Kazman, R., S.G. Woods, and J.S. Carrière, *Requirements for Integrating Software Architecture and Reengineering Models: CORUM II*, in *Working Conference on Reverse Engineering (WCRE 1998)*. 1998.
5. OMG. *Architecture-Driven Modernization (ADM)*. 2012; Available from: http://adm.omg.org/.
6. Sánchez Cuadrado, J., et al., *Parametrización de las transformaciones horizontales en el modelo de herradura*, in *Jornadas de Ingeniería de Software y Bases de Datos (JISBD'12)*. 2012: Almería, Spain.
7. Kavakli, E. and P. Loucopoulos, *Goal Driven Requirements Engineering: Evaluation of Current Methods*, in *Exploring Modelling Methods for Systems Analysis and Design (EMMSAD 2003)*. 2003: Klagenfurt/ Austria.
8. Etien, A. and C. Salinesi. *Managing requirements in a co-evolution context*. in *Requirements Engineering (RE 2005)*. 2005. Paris, France.
9. Herrmann, A., A. Wallnöfer, and B. Paech, *Specifying changes only - a case study on delta requirements*, in *REFSQ 2009*. 2009, Springer-Verlag: Essen, Germany. p. 45-58.
10. Salinesi, C., A. Etien, and I. Zoukar, *A systematic approach to express IS evolution requirements using gap modelling and symilarity modelling techniques*, in *International Conference on Advanced Information Systems Engineering (CAiSE'05)*. 2005: Porto, Portugal.
11. Etien, A. and C. Rolland, *Measuring the fitness relationship.* Requirements Engineering Journal, 2005(10): p. 184-197.
12. Rahn, E. and P.A. Bernstein, *A survey of approaches to automatic schema matching.* VLDB 2001. **10**: p. 334-350.
13. Madhavan, J., P.A. Bernstein, and E. Rahn, *Generic Schema Matching with Cupid*, in *27 th International Conference on Very Large Data Bases (VLDB 2001)*. 2001: Roma, Italy.
14. Bernstein, P.A. *Applying Model Management to Classical Meta Data Problems*. in *First Biennial Conference on Innovative Data Systems Research (CIDR 2003)*. 2003. Asilomar.
15. Wieringa, R., *Design Science as Nested Problem Solving*, in *4th International Conference In Design Science Research In Information System and Technology (DESRIST'09)*. 2009: Malvern, PA, USA.
16. Martin, M.J.C., *Managing Innovation and Entrepreneurship in Technology-Based Firms*, ed. I.T.O.E. MANAGEMENT. Vol. 43. 1996.

Theme Identification in RDF Graphs*

Hanane Ouksili

PRiSM, Univ. Versailles St Quentin, UMR CNRS 8144, Versailles France
hanane.ouksili@prism.uvsq.fr

Abstract. An increasing number of RDF datasets is published on the Web. A user willing to use these datasets will first have to explore them in order to determine which information is relevant for his specific needs. To facilitate this exploration, we present an approach allowing to provide a thematic view of a given RDF dataset, making it easier to target the relevant resources and properties. Our approach combines a density-based graph clustering algorithm with semantic criteria in order to identify clusters, each one corresponding to a theme. Prior to clustering, the initial RDF graph is simplified, and user preferences are mapped into a set of transformations applied to the graph. Once the clusters are identified, labels are extracted to express their semantics. In this paper, we describe the main features of our approach to generate a set of themes from an RDF dataset.

Keywords: Theme idendification, RDF(S) data, Clustering.

1 Introduction

An increasing number of RDF datasets is published on the Web, making a huge amount of data available for users and applications. In this context, a key issue for the users is to locate the relevant information for their specific needs. A typical way of exploring RDF datasets is the following: the users first select a URI, called a seed of interest, which they are willing to use as a starting point for their queries; then they explore all the URIs reachable from this seed by submitting queries to obtain information about the existing properties.

To facilitate this interaction, a thematic view of an RDF dataset can be given in order to guide the exploration process. We argue that once the data is presented as a set of themes, it is easier to target the relevant resources and properties by exploring the interesting topics only. In this paper, we present our approach for theme identification which combines a density-based graph clustering algorithm with semantic clustering criteria in order to identify clusters, each one corresponding to a theme.

The paper is organized as follows. Section 2 gives an overview of our proposal. Section 3 details the preprocessing step. Section 4 presents the clustering algorithm and we discuss methods of describing themes in Section 5. Our prototype and an example scenario are described in Section 6, related works are provided in Section 7, and finally, Section 8 concludes the paper.

* This work was supported by Electricity of France (EDF R&D).

2 General Principle of Theme Identification

Given an RDF dataset, our goal is to identify a set of themes and to extract the labels or tags which best capture their semantics. Providing this thematic view raises several questions:

- Which information could be used to define a theme?
- As different users may not have the same perception of the data, how to capture their preferences and use them for building the themes?
- Finally, once the themes have been identified, how to label them so as to make their semantic as clear as possible to the user?

Our approach relies on the idea that a theme corresponds to a highly connected area on the RDF graph. The more a set of resources is connected, the more likely it is that they belong to the same theme or are related to the same topic. We will therefore use the structure of the RDF graph itself in order to build the themes. We apply a graph clustering algorithm which identifies these highly connected areas and their neighborhood in order to form clusters, each one corresponding to a theme.

The structure of the graph alone is not sufficient to provide meaningful themes. Indeed, different users may have distinct perceptions of what a theme is. If we consider a dataset providing information about universities and scientists, one possible view is that themes correspond to research areas such as Mathematics or Physics, another one is that themes correspond to research teams located in the same geographical area. These preferences will be used for identifying the themes, in addition to the structure of the graph.

User preferences are captured by specifying the characteristics of all resources which should be assigned to the same cluster (for example, resources having the same value or linked by a given property). Each preference will be mapped into one or several transformations applied to the graph. For example, if the user expresses that two resources related by the *owl:sameAs* property should be assigned to the same cluster, the transformation will consist in merging the corresponding nodes in the graph.

An overview of our approach is given in Figure 1. It comprises three main steps, (i) preprocessing, where transformations are applied on the RDF graph, (ii) graph clustering, where themes are identified, and (iii) label extraction which provides a summary of the content of each cluster. In this paper, we mainly address the first two steps.

3 Preprocessing

The initial RDF graph will be transformed prior to the execution of the clustering algorithm. Some transformations are systematic regardless of the context, others consist in integrating user preferences in the graph. This section describes both of them.

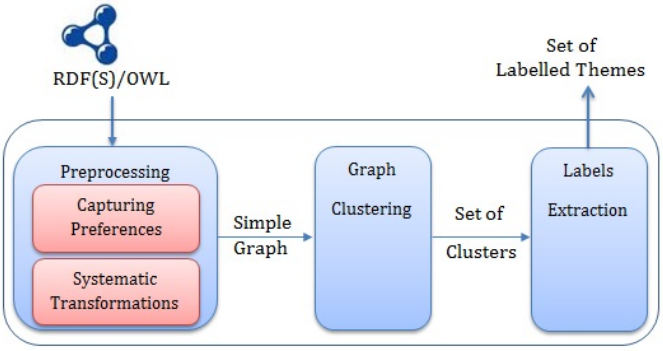

Fig. 1. Overview of our Approach

3.1 Systematic Transformations

Some of the information in the initial graph is not useful in order to group the nodes into meaningful clusters.

In the initial RDF graph, edges are oriented and labeled with the name of a predicate. The clustering algorithm used in our approach will try to identify highly connected areas, regardless of the orientation of the edges; no matter what the orientation of an edge is, what we are interested in is that some semantic relation exists between the resources. For example, consider *dbo:influenced* property that is asymmetric in nature; if we have the triplet $\langle r_i \ \textit{dbo:influenced} \ r_j \rangle$. We are not interested in which researcher between r_i and r_j that influenced the other; the most important is there is a semantic relationship between the two researchers. We can therefore simplify the graph by removing the orientation of the edges. Similarly, the clustering algorithm will not use the label of the edges, and they are also removed from the graph.

An RDF graph contains several types of nodes which can be either resources or literals. A literal is related to one resource and is a characteristic of this resource. Obviously, a resource and the related literals should be grouped into the same cluster. We could therefore apply the clustering algorithm on a simplified version of the graph which doesn't contain the literals.

The output of the preprocessing stage is a graph where the labels, orientation of the edges and literal nodes have been removed. Figure 2 shows an example of simplified graph (2(b)) corresponding to an RDF dataset (2(a)).

3.2 Capturing User Preferences

As stated earlier, the structure of the graph alone is not sufficient for the identification of meaningful clusters. Sometimes the density of the graph doesn't fully capture semantic closeness: for example, two resources might not be located in a very connected area of the graph, but if there is an edge in the graph relating them, and if this edge expresses a strong semantic link (e.g. *owl:sameAs*), the

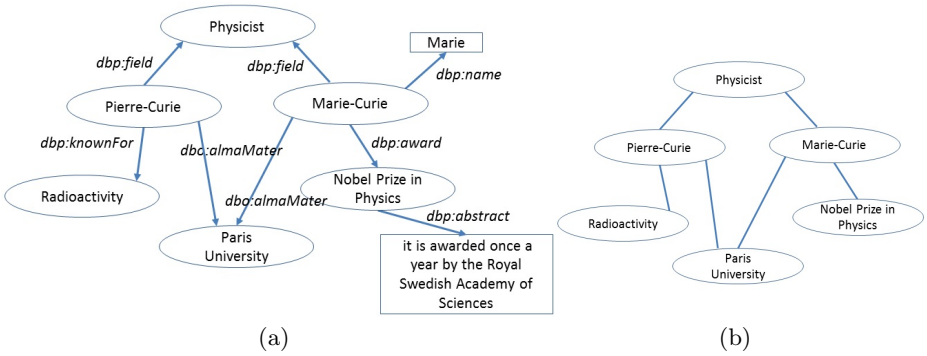

Fig. 2. Transformation of the Initial RDF Graph

two resources should be assigned to the same cluster. Furthermore, for the same dataset, different users might have different points of view and be interested in distinct properties. To capture this need, the clustering should take into account these properties as semantic criteria.

In our approach, user preferences are captured by mapping each of them into one or several graph transformation primitives. We consider that there are mainly two kinds of preferences a user might want to express. The first one is that two resources related by a given property should belong to the same theme. The second one is that a set of resources having the same value for a given property should belong to the same theme.

Grouping Two Resources According to a Property. Some properties express a strong semantic link that should be used as a clustering criteria. For example, resources linked by the *owl:sameAs* property should obviously be assigned to the same cluster, and this is true for any user in any context. Besides, some users may wish to give a property more importance than other users. For example, if we consider a dataset containing information about scientists in different research domains, a given user might consider that the resources *Student* and *Scientist* related by the *dbo:doctoralAdvisor* property should be grouped in the same cluster. This kind of preference is taken into account by merging the two resources.

Grouping a Set of Resources According to a Property. Resources that should be assigned to the same theme are not always linked by a property; the semantic closeness between them can be expressed by the values of some shared property. In other words, a set of resources having the same value for a property p should be assigned to the same cluster. For instance, the user could state that scientists having the same value for the *dbo:field* property should be in the same cluster, thus ensuring that scientists of the same research domain are grouped together. This kind of preference is taken into account by creating in the graph a highly connected area containing the specified resources. If we consider the set R of

resources r_i having the same value for a property p, then an edge (r_i, r_j) will be added for each pair (r_i, r_j) of resources such that r_i and r_j are in R, unless the edge already exists in the graph.

4 Clustering Algorithm

The clustering algorithm at the core of our approach has to fulfill a set of requirements, the first of which is exploiting the density of the graph to enable the identification of clusters corresponding to highly connected areas of the graph. The second requirement is that the algorithm should not require the number of clusters as a parameter, as this information cannot be known prior to clustering in our context. Finally, resulting clusters provided by the algorithm should not necessarily be disjoint, as it is possible that two distinct resources in our initial graph belong to two different themes.

We have chosen the algorithm proposed by [1] and initially used in the domain of bioinformatics. It is a density-based algorithm producing possibly overlapping clusters.

The algorithm operates in three steps. First (i), it computes the weights of each node in the graph using the concept of k-core. Consider that the degree of a node is the number of his adjacent nodes. A k-core is a graph in which the minimal node degree is k. The weight of a node S_i is computed based on the highest possible k-core value in the subgraph composed of S_i and its adjacent nodes; once the weights have been computed, (ii) the nodes are explored in a descending order of their weights; each node S_i will initiate a cluster, and for each adjacent node S_j such as the difference between the weights of S_i and S_j is below a threshold t is assigned to the same cluster as S_i; finally, (iii) once all the nodes have been explored, the algorithm enriches the clustering by checking all the adjacent nodes for a given cluster; if for a node S_i in a cluster C_i, the subgraph composed of S_i and its adjacent nodes is highly connected, then all the adjacent nodes of S_i will also be added to C_i. This will enable nodes to be part of more than one clusters.

5 Labels Extraction

Goal of this step is to provide the user a view of the cluster content by extracting a set of relevant labels that describe the theme. The set of labels is extracted from the names of RDF resources is composed by the top-k keywords having the high weight in the cluster C_i. The weight w_{ij} of the keyword j (noted $keyword_j$) in the cluster C_i is computed according to the degree of the node j (noted $node_j$). We note that $keyword_j$ appears in the name of $node_j$. We give an example in Figure 4, where selected theme represents a set of researchers workings in the field of physics. The top-1 labels extracted using our approach is *"Physics"* as we can see in the name of the sub window of the figure. This label reflects the semantic content of the cluster. We can add more labels by increasing the value of k.

This approach can be extended to use more characteristics to calculate the weight of keyword by combining the degree of the node with the frequency of the keyword in the cluster. Castano et al. [2] use the most frequent keyword combining with the most frequent type of entities in the cluster. Another alternative would be to use an adaptation of the tf-idf function to determinate the weight w_ij. In this way, the relevant of the $keyword_j$ is proportional to its frequency of the keyword in the cluster C_i and its scarcity in other clusters C_k with $k \neq i$.

6 Our System

We have implemented a tool to support our approach for theme identification. The system requires two types of parameters: (i) clustering parameters, used to specify thresholds for assigning a node to a cluster, and (ii) semantic parameters, used to capture user preferences.

To illustrate the way our tool is used for theme identification, consider the following example of an RDF dataset extracted from DBPedia (see Figure 3). This dataset contains resources describing scientists working in different domains with their organizations and their countries. Assume that the user wants to

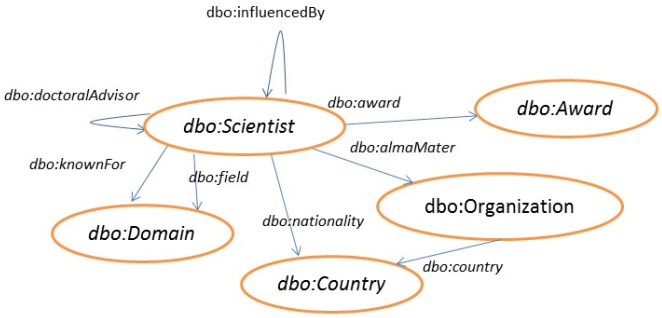

Fig. 3. Description of the RDF Dataset

identify themes in the input graph, and would like scientists from the same domain to be assigned to the same cluster. As the research domain is represented by the *dbo:field* property in our example, the user will indicate that two resources having the same value for this property should be assigned to the same cluster. He can repeat the clustering process either on the initial graph by adding new semantic parameters, or on a cluster obtained in previous iterations in order to get further details.

According to the preference set by the user, scientists of the same domain will be assigned to the same cluster. But it may happen that this property is not defined for some of the scientists in the dataset, and the user would therefore like to use another semantic criteria. For example, he could state that scientists related by the *dbo:doctoralAdvisor* property should be assigned to the same cluster.

Figure 4 shows the user interface of the system. The list of clusters is displayed on the left side and the initial RDF graph on the right side. The cluster selected in the list can be highlighted on the graph (green nodes) or opened as a new RDF graph. In Figure 4, the selected cluster represents the field of *Physics*.

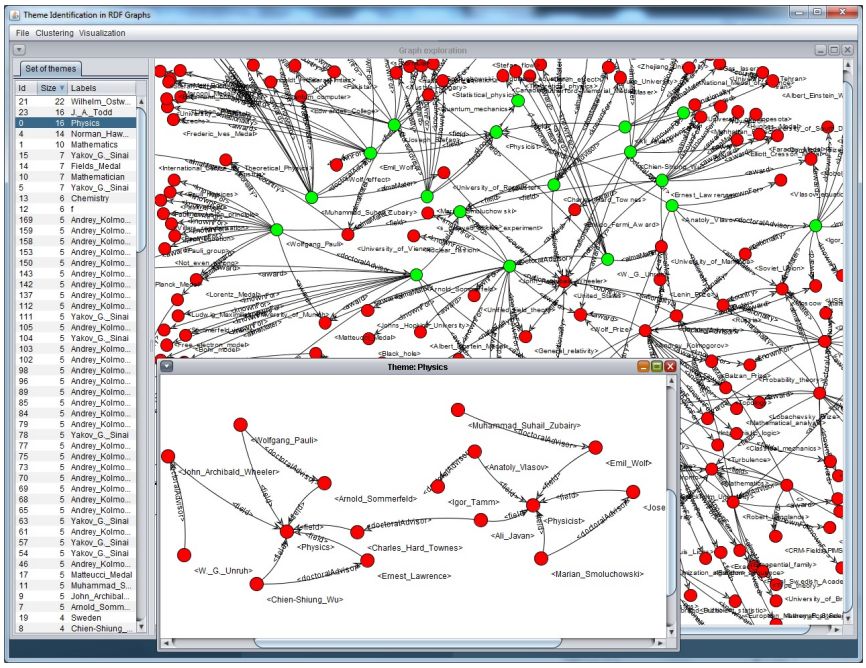

Fig. 4. Visualization of the themes

7 Related Works

Theme identification approaches have been proposed for text documents [8] or for other types of data published on the web, e.g. social networks [3], youtube documents [6] and DBpedia [7]. The goal of these approaches is to facilitate the search process and the navigation into the dataset. All of them use clustering techniques. Unlike our approach, they do not consider RDF datasets except the one described in [7]. Furthermore, they rely on text comparison to compute the distance between documents. This distance is used as the similarity measure for the clustering algorithm.

Despite the increasing amount of RDF(S)/OWL datasets available online, the problem of discovering themes have received little attention. Some works have focused on improving the quality of data by grouping resources to detect concepts and induce new classes or refine existing one [4, 5].

The closest work to ours is an approach for topic identification presented in [2]. It exploits a graph generated from an input RDF dataset, by adding new

edges between resources that have an important number of similar terms in their labels. A clustering algorithm is then applied to identify regions that are highly connected in the graph, which represent the topics. Similarly to our approach, this work is based on a clustering algorithm, but focuses only on identifying highly connected areas while we combine the density-based clustering process with semantic criteria capturing user preferences.

8 Conclusions

In this paper, we have proposed an approach for theme identification in RDF datasets. It combines a density-based clustering algorithm and semantic criteria capturing user preferences. Our approach comprises three stages: (1) preprocessing and capturing user preferences, (2) density-based clustering to form the clusters and (3) extraction of labels to describe the semantic of the cluster. Preprocessing consists mainly in simplifying the graph and removing the information which is not necessary for the clustering algorithm. Users' preferences are captured by mapping them into graph transformation primitives. Our approach differs from existing ones such as [2] in that it combines structural and semantic criteria for graph clustering. We have implemented a system for theme identification. Future works include the extension of the approach by improving label identification and providing the user with a summary of the clusters' content to describe its semantics. We are currently experimenting the use of our system on different RDF datasets in order to evaluate the precision of the clustering and the performances of the system.

References

1. G. Bader and C. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC bioinformatics*, 27:1–27, 2003.
2. S. Castano, A. Ferrara, and S. Montanelli. Thematic clustering and exploration of linked data. *Search Computing*, pages 157–175, 2012.
3. S. Castano, A. Ferrara, and S. Montanelli. Mining topic clouds from social data. *In Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems (MEDES '13)*, pages 108–112, 2013.
4. K. Christodoulou, N. W. Paton, and A. A. A. Fernandes. Structure inference for linked data sources using clustering. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops on - EDBT '13*, page 60. ACM Press, 2013.
5. N. Fanizzi, C. DAmato, and F. Esposito. Metric-based stochastic conceptual clustering for ontologies. *Information Systems*, 34(8):792–806, Dec. 2009.
6. U. Gargi, W. Lu, V. Mirrokni, and S. Yoon. Large-Scale Community Detection on YouTube for Topic Discovery and Exploration. *ICWSM*, pages 486–489, 2011.
7. R. Mirizzi and A. Ragone. Semantic wonder cloud: exploratory search in DBpedia. In *ICWE 2nd Int. Workshop on Semantic Web Information Management (SWIM 2010)*, pages 138–149, 2010.
8. H. Shahsavand Baghdadi and B. Ranaivo-Malançon. An Automatic Topic Identification Algorithm. *Journal of Computer Science*, 7(9):1363–1367, 2011.

Information Retrieval Framework based on Social Document Profile

Amna Dridi
supervised by: Mouna Kacimi

Faculty of Computer Science
Free University of Bozen-Bolzano I-39100, Italy
{Amna.Dridi, Mouna.Kacimi}@unibz.it

Abstract. Social networks provide rich information about user interests and activities representing a valuable source for search personalization. However, social information is typically large and dynamic making its exploitation to obtain relevant search results a very challenging task. This work presents a PhD project plan that investigates Social Information Retrieval. The goal is threefolds: (1) create confidence area for information search by community detection based on tags similarity (2) introduce a new notion of Social Document Profile based on user activities, and (3) propose a novel ranking model based on social relevance.

1 Introduction

1.1 Motivation

Social networks are becoming one of the predominant sources of information. Users of such networks publish documents that can take different forms, including text, image, audio, and video. Additionally, they can perform different types of actions around published documents. These actions can be classified as *descriptive* or *reactive*. Descriptive actions, mainly *tagging*, reflect the content of documents, while reactive actions such as *like*, *dislike*, *rate*, *favorite*, *share*, and *comment* reflect users' feedbacks regarding documents. This rich repository of users' actions triggered many research works to exploit social information for search personalization [3–5, 5, 5, 10, 12–14]. Most of the existing techniques consider descriptive actions (tagging) as the main indicator of users interests and thus use them for building users and documents profiles. However, relying only on tagging actions to provide relevant search results to users' needs is not sufficient. For example, a video tagged by {*Wolswagen*, *car*, *advert*} would be returned as a relevant result to the query *"car advert"* initiated by a user interested in *"Wolswagen"*. Knowing that the video features people speaking in fake Jamaican accents, some users would find it funny while some others would find it offensive. In this case, the video should be relevant only if it is liked by users having similar profiles to the query initiator. Consequently, the pool of users' reactions should be exploited to refine the search space and give a new definition for social document relevance. The contrast between descriptive actions

which are directly related to the content of documents and reactive actions that show users' personal preferences makes the exploitation of social information a challenging task.

1.2 Contribution and Paper structure

We propose to provide tailored answers to users' needs by exploiting social information in two different stages. First, we use descriptive actions to create, for each user, a confidence search area according to his profile. Second, we use both descriptive and reactive actions to define a social profile, per confidence area, for each document. The novel contribution by this paper has the following salient properties:

1. We model a social information retrieval framework as an undirected graph of social entities (User, Document, Tags and Clicks) where links represent entities relations generated in a social context, Tags represent descriptive actions, and Clicks represent reactive actions.
2. We exploit user profile as a tool for community detection based on Tags similarity. The goal is to establish a confidence search area for each user.
3. We propose a novel Social Document Profile based on a tripartite graph (Content, Tags, Clicks) that represents documents not only using their content but also their social profile given by Tags and Clicks.
4. We propose a novel scoring model that combines content relevance based on user profile and social relevance based on social document profile.

Our proposed approach goes beyond existing IR personalization techniques in several ways. First, it combines two areas: community detection in social networks and information retrieval. Second, unlike existing approaches, we define personalization approach based not only on user profile but also on document social profile. Third, none of the existing approaches takes into account clicks as social information defining document profile.

2 Related Work

Search personalization using social information has been investigated extensively. The first class of approaches limits social information to annotations or tags [3–5, 13]. For instance, Bouadjenek et. al., [4] use tags to build user profiles and then use those profiles for query expansion. The idea is to compute social proximity between each query and the profile of its initiator. Vellet et. al., [13] present two techniques that build user and document profiles. The first technique use a vector space model incorporating the concepts of tag inverse document frequency and tag inverse user frequency in folksonomy systems. By contrast, the second technique adapts the BM25 probabilistic model to user and document vectors. Similarly, Bouadjenek et. al., [3] propose a framework for social web search, called LAICOS, which construct document profiles based on their content and associated tags. Cai et. al., [5] examine the limitations of TF-IDF-based

models showing that using absolute term frequency favors active users against non-active users. Moreover, inverted document frequency is not necessary useful in indicating users' preferences on tags or how a document is relevant to tags. Thus, the authors use a Normalized Term Frequency (NTF) to indicate the preference degree of a user on a tag and thus construct user profile. Then, they perform search by matching users' profile and documents profile.

The second class of approaches exploits, in addition to tags, social relationships between users [1,6,9,10,12]. For instance, Carmel et. al., [6] re-rank search results based on friendship relationships among users. Schenkel et. al., [10] propose a top-k algorithm for social search and ranking with two dimensional expansions: semantic expansion that considers the relatedness of different tags and social expansion that considers the strength of relations among users. In the same context, Gou et al. [9] propose a framework called SNDocRank that considers documents content and the relationship between information seekers and documents owners by combining TF-IDF and Multi-level Actor Similarity (MAS) algorithm. Tang et. al., [12] selects the closest sub topics to the query and then looks for the most influential users. They have developed an influence maximization algorithm to find the sub network that closely connects influential users. Similarly, Ben Jabeur et. al., [1] define social scores based on users' relationships which depend on users' positions in the social network and their mutual collaborations.

All approaches described above focus on how to generate user profile using social information but none of them takes into account social document profile. In our work, we exploit user profile not at query time but to detect interest communities as confidence search areas. Moreover, we build a social document profile based on clicks which was not considered in related work. A work that went beyond using only tags and user relationships is by Wang et. al., [14] who define users' interests based on users' activities. However, the authors consider activities that are not related to documents but about social relationships such as subscription to groups. In our work, we use Clicks which are main indicators of documents social relevance.

Another research area related to our work is community detection where various methods have been proposed [2, 6, 7]. For instance, Bothorel et. al., [2] develop measures of centrality based on the shortest paths in social networks such as: Degree Centrality, Betweenness Centrality, and Closeness Centrality. De Meo et. al. [7] take a different approach than using network structure and propose Jaccard coefficient to calculate the similarity between users in Facebook based on social activities. In case of a null result, Jaccard coefficient has a disadvantage of the similarity lack between two users whereas this is not true. To solve this problem, a popular parameter introduced by social science called Katz coefficient is used to calculate the similarity between two users taking into account all possible paths between two nodes. Carmel et. al. [6] consider similarity between two individuals according to common activity in the context of LC's [1] social software: co-usage of the same tag, co-tagging of the same docu-

[1] IBM Lotus Connections

ment, co-membership of the same community, or co-commenting on the same blog entry. The latter approach fits our needs but since we do not have access to the corresponding platform, we adopt Katz coefficient and use it as tool for community detection in social networks because of its effectiveness to take into account various types of links between nodes in the social graph.

3 Social Information Retrieval Framework

We define the Social Graph SG as a tuple SG = {U, D, T, C, A_1, A_2} where U = {u_1, ..., u_k}, D = {d_1, ..., d_l}, T = {t_1, ..., t_m} and C = {c_1, ..., c_e} are respectively the set of Users, Documents, Tags and Clicks. $A_1 = \{u_i, d_j, t_f\} \in U \times D \times T$ is a set of annotations reflecting each user u_i tagging document d_j with tag t_f and $A_2 = \{u_i, d_j, c_r\} \in U \times D \times C$ is a set of clicks reflecting each user u_i reacting to document d_j using click c_r (see Figure 1).

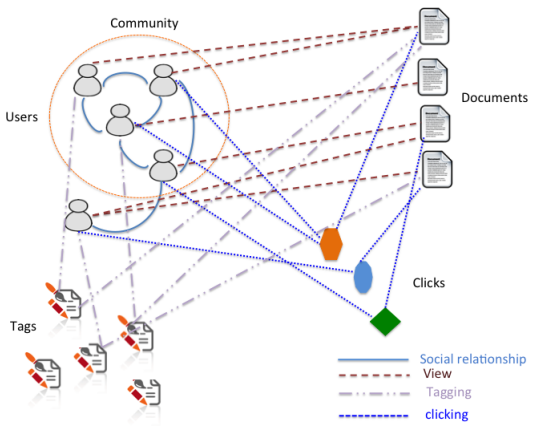

Fig. 1. Social Information Retrieval Graph

3.1 Overview

Our personalized search strategy consists in the following steps. First, we extract users' communities from social networks based on users' profiles. The profile of a user is defined by the set of tags he used to annotate documents. Thus, the community detection problem is reduced to computing tags similarity by using the subgraph G = (U, T) of the social graph SG. Second, upon receiving a search query $Q = \{q_1, ..., q_n\}$ from a user u, we proceed as follows:

1. We retrieve the topk relevant results to the query. Each result is associated with a content relevance score; the more relevant and important a result is with respect to the query, the higher its relevance score.

2. For each of the topk results, we compute its social score based on how popular it is in user u's community. This popularity is defined by related clicks (share, favourite, comment, etc).
3. The results are then re-ranked based on the combination of the content relevance score and the social relevance score of each result.

3.2 Social User Profile-based community detection

Social User Profile Our proposed model for social information retrieval is based on a central phase of community detection. Our aim is to detect community of interest to personalize IR processes. We propose to use the subgraph G = (U, T) of the social graph SG to detect similar users based on the tags they use. Note that, we take into account the time factor s since users' interest change over time. Therefore, the social user profile P_i of user u_i is defined by $P_i = \{t_1, \ldots, t_m\}^s$. To detect community between users it is then to compute tags similarity.

Community Detection We propose to adopt Katz coefficient for community detection. Katz coefficient is a similarity index proposed in the field of social science and was recently rediscovered in the context of collaborative recommendation and Kernel methods where they are known as Von Neuman Kernel. Katz proposed a method of calculating similarity taking into account not only the number of direct links between elements, but also the number of indirect links [8].

$$Katz := \sum_{l=1}^{N} \beta^l {paths^l}_{i,j}$$

where l is the length of the path and β^l is the appropriate weight to path l.

3.3 Social Document Profile

Each document has a social profile defined by annotations (Tags) and Clicks in addition to its content. Therefore, a document D is defined by the threefold {Ct, T, C} where Ct, T and C respectively correspond to Content, Tag and Click. Therefore, a document is evaluated through two measures: *content relevance* and *social relevance*.

Content relevance. To compute the relevance of a document d_x to user query, we use BM25 (or Okapi) scoring function given by :

$$BM25(d_x, q_i) = IDF(q_i).\frac{f(q_i, d_x).(k_1 + 1)}{f(q_i, d_x) + k_1.(1 - b + b.\frac{|d_x|}{avgdl})}$$

where f(q_i, d_x) is the count of term q_i in document d_x, $|d_x|$ is the length of document d_x, avgdl is the average document length in the collection of documents,

$k_1 = 1.2$ and b = 0.75, IDF(q_i) is the inverse document frequency weight of the query term q_i which is computed as :

$$IDF(q_i) = log\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}$$

where N is the total number of documents in the collection, and n(q_i) is the number of documents containing n(q_i). Thus, the content relevance score of a document x is given by:

$$Rel(d_x, Q) = \sum_{i=1}^{n} BM25(d_x, q_i)$$

Social relevance. To compute *social relevance*, we use the tripartite graph (User, Document, Click) from the Social Graph SG. We consider the Clicks C = $\{c_1, \ldots, c_e\}$ to estimate the social popularity of a document in a given community. For the same query by two different users returned results are ordered differently depending on the social context of each user. Our idea for the social relevance computation is to to find a social score for clicks which is the weighted sum of clicks weighted scores. We consider the following click score of document d_x clicked by click c_i in the community of user u:

$$cs(d_x, c_i, u) = \frac{count(c_i, d_x, u)}{count(d_x, u)}$$

where: $count(c_i, d_x, u)$ is the number of users, in the community of user u, who used click c_i for document d_x, and $count(d_x, u)$ is the total number of users, in the community of user u, who clicked on document d_x. By combining the click scores, we obtain the social score of document d_x in the community of user u given by:

$$SS(d_x, u) = \sum_{i=1}^{e} \alpha_i cs(d_x, c_i, u)$$

where e is the number of clicks types (For example, in Facebook we have e=3 because we have 3 clicks types : like, share and comment) and $\sum_{i=1}^{n} \alpha_i = 1$ where α_i is a weighted coefficient selected by the query initiator.

3.4 Social Ranking Function

We use a linear combination of the content score $Rel(d_x, Q)$ and the social score $ss(d_x, u)$ to obtain the final score of a document d_x returned as a result for query Q initiated by user u:

$$S(d_x, u) = \lambda Rel(d_x, Q) + (1 - \lambda)SS(d_x, u)$$

where $0 \leq \lambda \leq 1$

4 Research Plan and Conclusion

As a short term objective, we plan to implement our personalized search approach and perform experiments on real-world data to evaluate its performance focusing on the following tasks: .

1. Compare our click-based personalization with tag-based personalization
2. Study closely the impact of the social document model on search results
3. Analyze how our technique performs depending on the level of activities in different communities.

4.1 Experimental Data

We will test our personalized search approach using data crawled from YouTube [2] which has the main characteristics needed for our solution. This dataset have been crawled during the period between October, 15th, 2012 and December, 25th, 2012. It contains 890682 videos, 282074 users and 1014190 information about social clicks (comment, favourite and rated).

Table 1. Statistical characteristics of YouTube dataset

Users	282074
Videos	890682
Clicks	1014190

4.2 Research Plan.

Our long term objectives consist in the following:

1. Investigate new techniques for community detection that go beyond tag similarity by involving users' reactions to published documents in social networks. We believe that building confidence search areas based on what users think about documents is a promising direction towards satisfying user's needs.
2. Extend the notion of document social relevance by considering not only positive feedbacks but also negative ones. The idea is to boost documents social scores if they receive positive feedbacks and penalize them otherwise. This task involve mainly mining users' comments to understand their interests and derive their judgment about published documents.
3. Develop an efficient and scalable ranking algorithm that can handle the fast growth of communities and the very high rate of content production together with tagging and clicking actions.

[2] www.youtube.com

4. Validate our proposed techniques using real datasets from social networks. We aim at investigating networks with different properties such as, Facebook, Twitter, and Delicious to understand the behavior of our approach is different environments.

Acknowledgements. This research was supported by the RARE project at KRDB research centre for knowledge and data at Free University of Bozen-Bolzano.

References

1. Ben Jabeur, L., Tamine, L., Boughanem, M.: A social model for Literature Access: Towards a weighted social network of authors. In Proceeding of RIAO 2010, pp. 32-39
2. Bothorel, C.: Social network analysis and unpopular content recommendation. Review of New Information Technologies (RNIT) 2011, Vol. A.5
3. Bouadjenek, M.R., Hacid, H., Bouzeghoub, M.: LAICOS: An open source platform for personalised social web search. In Proceeding of KDD 2013, pp. 1446-1449
4. Bouadjenek, M.R., Hacid, H., Bouzeghoub, M., Daigremont, J.: Personalized social query expansion using social bookmarking systems. In Proceeding of SIGIR 2011, pp. 1113-1114
5. Cai, Y., Li, Q.: Personalized Search by Tag-based User Profile and Resource Profile in Collaborative Tagging Systems. In Proceedings of CIKM 2010, pp. 969–978
6. Carmel, D., Zwerdling, N., Guy, I., Ofek-Koifman, S., Har'el, N., Ronen, I., Uziel, E., Yogev, S., Chernov, S.: Personalized Social Search Based on the User's Social Network. Proceedings of CIKM 2009, pp. 1227-1236
7. De Meo, P., Ferrara, E., Fiumara, G.: Finding Similar Users in Facebook, Social Neworking and Community Behavior Modeling: Qualitative and Quantitative Measurement. IGI Global 2011, pp. 304-323
8. Fouss, F., Pirotte, A., Renders, J.M., Saerens, M.: Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation. In Proceeding of TKDE 2006, Vol.19, pp. 2007
9. Gou, L., Zhang, X.L., Chen, H.H., Kim, J.H., Giles, C.L.: Social Network Document Ranking. Proceedings of JDCL 2010, pp. 313-322
10. Schenkel, R., Crecelius, T., Kacimi, M., Michel, S., Neumann, T., Parreira, J.X., Weikum, G.: Efficient Top-k Querying over Social-tagging Networks. In Proceedings of SIGIR 2008, pp. 523-530
11. Ronen, I., Shahar, E., Ur, S., Uziel, E., Yogev, S., Zwerdling, N., Carmel, D., Guy, I., Har'El, N., Ofek-Koifman, Sh.: Social Networks and Discovery in the Entreprise (SaND). In Proceedings of SIGIR 2009, pp. 836-836
12. Tang, J., Wu, S., Gao, B., Wan, Y.: Topic-level Social Network Search. In Proceedings of KDD 2011, pp. 769-772
13. Vallet, D., Cantador, I., Joemon, M.J.: Personalizing Web Search with Folksonomy-based User and Document Profiles. In Proceedings of ECIR 2010, pp. 420-431
14. Wang, Q., Jin, H.: Exploring Online Social Activities for Adaptive Search Personalization. In Proceedings of CIKM 2010, pp. 999-1008