DOI 10.1007/s00165-015-0337-z
BCS © 2015
Formal Aspects of Computing (2015) 27: 867–884

Formal Aspects
of Computing

Finding semantic associations in hierarchically structured groups of Web data

Domenico Rosaci[1]

[1] DIIES Department, University of Reggio Calabria, Via Graziella Loc, Feo di Vito, 89122, Reggio Calabria, Italy

Abstract. Most of the activities usually performed by Web users are today effectively supported by using appropriate metadata that make the Web practically readable by software agents operating as users' assistants. While the original use of metadata mostly focused on improving queries on Web knowledge bases, as in the case of SPARQL-based applications on RDF data, other approaches have been proposed to exploit the semantic information contained in metadata for performing more sophisticated knowledge discovery tasks. Finding semantic associations between Web data seems a promising framework in this context, since it allows that novel, potentially interesting information can emerge by the Web's sea, deeply exploiting the semantic relationships represented by metadata. However, the approaches for finding semantic associations proposed in the past do not seem to consider how Web entities are logically collected into groups, that often have a complex hierarchical structure. In this paper, we focus on the importance of taking into account this additional information, and we propose an approach for finding semantic associations which would not emerge without considering the structure of the data groups. Our approach is based on the introduction of a new metadata model, that is an extension of the direct, labelled graph allowing the possibility to have nodes with a hierarchical structure. To evaluate our approach, we have implemented it on the top of an existing recommender system for Web users, experimentally analyzing the introduced advantages in terms of effectiveness of the recommendation activity.

Keywords: Semantic Web, Metadata model, Semantic searching algorithms, Recommender systems

1. Introduction

As widely emphasized in the W3C specifications, the Semantic Web [Sem09] is a Web of data, having the main purpose of providing the possibility to share and reuse data across different applications. To this purpose, it is necessary to use some model for representing how the data relate to actual real world entities, and for expressing the existing relationships among data. A model having this characteristic can be defined a metadata model, since the data relationships represented in it provide a semantic description of the data (a metadata), specifying what a data means rather than only its value. These are the characteristics of the resource description framework (RDF) [Rdf13], that has been proposed by the W3C as the data representation model on which the Semantic Web is based.

Correspondence and offprint requests to: D. Rosaci, e-mail: domenico.rosaci@unirc.it

This model allows to express statements about resources, using expressions of the form subject-predicate-object [Hje01]. The subject and the object are resources, and the predicate represents a relationship between the subject and the object.

For instance, consider a resource, identified as "Domenico Rosaci", consisting of the URL www.domenicorosaci.it/index.html, and the resource, identified as "rosaci-email", consisting of the email `domenico.rosaci@unirc.it`. In RDF we can express the statement "Domenico Rosaci has the email address rosaci-email" by a directed edge between the nodes "Domenico Rosaci" and "rosaci-email", where the edge represents the predicate "has the email". In other words, each set of RDF statements can be represented as a labelled, directed graph, accordingly with the intrinsical graph-structure of the Semantic Web, whose properties have been widely analyzed [TTK08].

The introduction of metadata has opened new possibilities to suitably exploit the information contained on the Web [WBB08]. Mainly, it is possible to use information software agents to handle and process available data, thanks to the machine readability provided by the metadata, thus giving the possibility to make automatic activities that the users manually performed in the past [KYK03].

However, the existing Web applications that use metadata for supporting user activities are mostly querying tools, based on languages suitable to handle RDF data. For example, the language SPARQL allows to express queries across diverse RDF data sources, and a number of approaches have been recently proposed in this setting [GGE09, ScS08, SSB08, KoJ07]. Some of these approaches aim at optimizing SPARQL queries on RDF data [GGE09, SSB08], analogously to that proposed for SQL on relational data. Others approaches show how it is possible to discover semantic associations [AnS03] between entities on RDF knowledge bases [KoJ07, Bar04, AMS05], that is how to find paths of possibly unknown length that connect the given entities and have a specific semantics. Viewed in the context of a graph model such as that for RDF, semantic associations represent certain graph signatures, as directed or undirected paths between entities, or subgraphs.

However, observing the characteristics of the information actually stored in the Web, as well as the features of the main Web applications, we can recognize that an important issue to be investigated is that of discovering semantic relationships between *groups of data*.

Indeed, in most of the Web activities, as in e-commerce, e-learning, e-government, social networks and so on, data are often grouped into collections, for representing data categories. This corresponds to the actual categorization of entities and resources to which data are associated as, for instance, groups of products in e-commerce or groups of users in social networks. These groups are often mutually related, or related to some single object. As widely recognized, many knowledge bases of interest today are best described as a linked collection of interrelated objects [ToF06, GeD05].

As an example, consider the case of a social network in which the users discuss about literature, and suppose that in this network there are several groups of discussion, e.g. *Italian literature*, *English literature*, *Spanish literature* and so on, where each group contains a given number of users. In this context, it is possible to conceive the statement "the user John is interested in contacting all the persons of the group Italian literature". Such a statement is composed by three logical terms: the subject "the user John", the predicate "is interested in contacting" and the object "all the persons of the group Italian literature". In this case, while the subject is a single entity, the object is a group of entities and therefore the predicate represents a relationship between the subject John and all the entities belonging to the group Italian literature. As another example, it is possible to imagine the statement "all the users of the group Italian literature are interested in contacting Umberto Eco". In this case the subject of the statement is a group while the object is a single entity, and therefore the predicate relates many entities to only one. Finally, we can also suppose to express the statement "all the users of the group Italian literature are interested in contacting all the users of the group English literature". In this case, the predicate relates a group of entities to another group of entities. The necessity to express this kind of statements, where groups of entities are involved, is very common in Web applications.

Resource description framework allows the representation of groups of data via suitable entities as, for examples, bags and sequences, therefore it is possible to represent in this framework relationships between groups of data. However, the question that we pose here is "how is it possible to exploit semantic relationships between groups of data to discover new, potentially useful, information?"

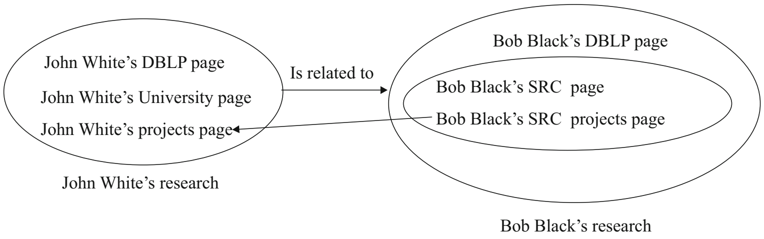


Fig. 1. An example of semantic relationship between groups

We remark that, to the best of our knowledge, no attempt to answer the question above has been done. Some techniques, as that proposed in [KoJ07], are capable of finding associations between entities in RDF knowledge bases, but they are not easily extendable to find associations in presence of groups of entities. Indeed, these techniques are based on the graph-structure of semantic data used by RDF, that allows to model a relationship between two entities as an arc between two nodes, each of which represents one of those entities. But if we suppose that some node of the RDF graph represents a group of entities instead of a single entity, it is possible that there exists some relationship between entities, or between an entity and a group, or between two groups, without the explicit presence in the graph of a path between two corresponding nodes. For example, consider the situation depicted in Fig. 1, that represents the statement "John White's research is related to Bob Black's research" by means of a directed arc between two nodes representing John White's research and Bob Black's research, respectively. In this situation, we suppose that John White's research is actually a group of entities, namely the Web pages dealing with the John White's DBLP publications, projects and personal page at his university. Also Bob Black's research is a group of entities, containing the Bob Black's DBLP page and a sub-group of entities related to the activities at the semantic research corporation (SRC). Moreover, another statement is represented in Fig. 1, that is "The Bob Black's SRC projects page is related to the John White's projects page". Now, suppose that we are interested to find entities that are mutually related.

The arc between the two nodes "John White's research" and "Bob Black's research" means that all the entities of John White's research are related to all the entities of Bob Black's research. This means, as a particular case, that the entity representing the "John White's projects page" is related to the entity representing the "Bob Black's SRC projects page". Since the arc between "Bob Black's SRC projects page" and "John White's projects page" means that the inverse relationship exists, we can conclude that "Bob Black's SRC projects page" and "John White's projects page" can be mutually associated.

However, we do not find a pair of arcs between the nodes associated to those two entities, and this makes not explicit the semantic relationship, that can be discovered only considering the group structure of the nodes. In other words, we highlight that: (i) it is possible to find semantic associations in presence of groups of entities by means of algorithms that consider both the direct relationships and the group structure; (ii) the simple graph structure is not suitable to support the design of these algorithms.

Indeed, we argue that to find associations between entities, or groups of entities, or between an entity and a group, it is necessary to determine sets of nodes that are mutually (semantically) connected. These nodes have to represent either single entities or groups of entities.

To this purpose, in this paper we propose a method to detect the associations above. The main idea underlying our proposal is that of designing a metamodel of the Web resources that, differently from the graph structure of RDF, can represent in a direct way both single entities and groups of entities, also allowing nesting of subgroups. This metamodel, called *framoid*, describes the semantic frame of the involved Web resources, and can be viewed as a collection of semantic sub-frames that we call *framels*. In other word, a framoid is a hierarchical structure, similar to that of a file system with files and directories, with the addition of the possibility to have relationships between the components of the framoids, i.e. the framels. A framoid can be also viewed as a generalization of a direct, labelled graph, in which the nodes can have a hierarchical structure.

Although this metamodel does not present any additional expressive power with respect to an RDF graph, makes it easy to represent semantic relationships in presence of groups of entities, without using the RDF containers. Indeed, if we use an RDF container to represent a group, in the case a semantic relationship involves that group it would be necessary to represent that the relationship involves all the elements of the container. In presence of a nesting structure in the groups, this representation would be obviously unsuitable.

We use the framoid metamodel to formalize our algorithm for detecting semantic associations in presence of groups of entities. Such an algorithm consists in determining particular sub-structures of the framoid, called *naturally emerging framels* (NE-framels), that represent objects (either single entities or groups of entities) semantically connected. We show that determining NE-framels in a framoid consists of finding the strongly connected components of the framoid, where these components are a generalization of the strongly connected components in a direct, labelled graph.

We highlight that the main assumption of our proposal supposes that a node has a structure semantically consistent. The foundation of such an assumption is that is can be considered sufficiently reasonable, mainly in a Web scenario. Our notion of framel represents the underlying actual situation of a group of objects. If an object is member of a group, it is reasonable that its semantics is consistent with that of the other members of the group, and with that of the group itself. In other words, each object of a group participates in the group, and this participation is a common, intrinsically semantic property of all the objects. For instance, if a group has two components as the list of *likes* and the list of *dislikes*, the consistency of the semantics does not derive from the exact meaning of the two components (that seem so different) but from the fact they represent the *likes* and *dislikes* of *that* group. If some other object establishes a relationship with that group, it is natural that it is establishing a relationship also with the *likes* and *unlikes* lists. For examples, if I'm interested in analysing the property of the group, it is very probable that I'm interested in analysing the lists of *likes* and *dislikes*. Roughly speaking, the unique assumption of semantic consistency we made in our framework is that each member of a framel represents something that actually belongs to the group represented by the framel.

In order to evaluate the suitability of finding semantic associations using our approach, we have implemented it on the top of a recommender system that supports the navigation of Web users. We argue that this kind of application represents a typical case in which determining semantic associations between Web resources can improve the effectiveness of the results. The recommender system MUADDIB (formerly MASHA) [RoS06, RSG09] is able to recommend its users with Web pages that should result of interest for them. To this purpose, it uses two well-known types of approaches, called *content-based* and *collaborative filtering*. We have added to the MUADDIB system the capability to also generate *semantic associations-based* recommendations, and we have performed some experiments on real users that show a considerable improvement of the effectiveness, in terms of some well-known evaluation measures.

We have also compared the effectiveness of the recommendations generated exploiting our framoid-based algorithm with that of the recommendations exploiting traditional, RDF graph-based semantic associations. This comparison has shown the significant advantage introduced by our approach with respect to the classical one, thanks to the additional information on the Web data structure available by using a framoid as a metadata model.

The plan of the paper is as follows. In Sect. 2 we present some related work. Section 3 presents the framoid metamodel, while Sect. 4 describes our approach to finding semantic associations based on that metamodel. Section 5 introduces a case study that highlights the possible practical usage of the approach. Section 6 presents the experiments we have performed to evaluate our approach and, finally, in Sect. 7 we draw our conclusion.

2. Related work

In the context of the emerging trends to extend traditional search engines to a more expressive semantic level [Wei09], the issue of discovering complex relationships in Semantic Web data has been investigated in several past works. These relationships are often called *semantic associations* [AnS03] and are generally represented by a path between two entities, or by a subgraph of an original graph of entities. For instance, in [AnS02] an approach that supports querying for semantic associations on the Semantic Web has been proposed, with the purpose to detect relationships between entities involving sequences of predicates, and sets of predicate sequences that interact in complex ways. This approach provides a suitable operator, called ρ operator, for expressing queries about such associations. Also in [Bar04], complex relationships are discussed and are referred to as semantic associations, and it is introduced a design of an indexing structure for the RDF graph that will make the discovery of the relationships described by these operators effective. Moreover, in [AMS05], the issue of how search results of semantic associations can be ranked is addressed. In [RMP05], the authors introduce heuristics for discovering a subgraph from simple paths towards more informative ones. In particular, this approach tries to discover what are the most relevant ways in which a given entity X is related to another entity Y, formalizing the response as a subgraph connecting X to Y. All these approaches, similarly to our one, have the purpose of exploiting metadata for improving and making machine readable the search of Web information. However, differently to our approach, they do not consider the presence of hierarchically structured groups of entities. Consequently, all the aforementioned approaches exploits a graph data model for knowledge representation, allowing the semantic associations search techniques to be built upon the graph algorithms for paths. Differently, our approach takes into account the presence of information about groups of entities, and exploits a novel metadata model for suitably representing such a kind of information. As a further difference, while the approaches above express semantic associations as paths between entities, our one detects groups of semantically related entities having a hierarchical structure, called *framels*.

Another type of approaches consider the presence of groups of entities in Web contexts, as in the case of social networks. For instance, in [AND08] an approach to discovering semantic associations between the reviewers and authors in a populated ontology is presented. This ontology was created by integrating entities and relationships from two social networks. As another example, in [Zhu09], the author describes a loosely coupled semantic data model, called SLN, able to semantically link resources and derive implicit semantic links according to a set of relational reasoning rules. The intrinsic relationship between semantic communities and the semantic space of SLN stands at the base of some proposed approaches to discovering reasoning-constraint, rule-constraint, and classification-constraint semantic communities. These last proposals consider, similarly to our approach, the existence of groups of entities in the structure of the involved virtual environment. However, they do not define a formal framework in which relationships between entities and/or group or entities can be explicitly represented to support the discovering of semantic associations, while our approach introduces such a framework.

The possibility to find semantic associations between groups of entities is considered in [GaR08], where the semantic associations are exploited to cluster agents having different personal ontologies. This approach is based on a meta-model that takes into account the explicit representation of groups of semantically related entities, as in our approach, but in that case the entities are agents while in our metamodel we represent groups of objects, possibly containing nested levels of sub-groups.

3. Framels and framoids

In this section, we describe the metamodel that we use in our approach. The basic notion that we introduce is that of *framel*. A framel represents a group of semantically related objects, possibly containing nested levels of subgroups. Its structure is similar of that of a directory in a file system, that can contain single objects (i.e. files) and subgroups (i.e. sub-directories). However, differently from the case of a file system, where a directory and a file are two conceptually distinct entities, the definition of a framel is completely recursive. A single object is considered as a framel (called *singleton*) and therefore a generic framel is defined as a set of composing framels. Formally:

Definition 3.1 (*Framel*) Let O be a set of objects. A *framel* on O is either (i) an object (that we also call a *singleton framel*) or (ii) a set of framels on O.

Example 3.2 (Framels) In Fig. 2a, a set O of objects is shown. Each of these object can be also regarded as a singleton framel. Figure 2b–h graphically depict some examples of framels on O. In particular, the Fig. 2b represents a framel $f1$ containing only the object a, i.e. $f1 = \{a\}$. The Fig. 2c represents a framel $f2$ containing as unique element an internal framel, which stores only the object a, i.e. $f2 = \{\{a\}\}$. This latter example highlights how a framel can be “encapsulated” an arbitrarily large number of times into external framels. The Fig. 2d represents a framel $f3$ containing the objects a, b and c, i.e. $f3 = \{a, b, c\}$. The Fig. 2e represents a framel $f4$ containing two elements, the first represented by the object a and the second which is in its turn a framel containing the objects b and c, i.e. $f4 = \{a, \{b, c\}\}$. The Fig. 2f represents a framel $f5$ containing, besides the same two elements of the framel $f4$ previously described, also a third element, which is in its turn a framel containing both the object d and a framel composed by the objects e and f, i.e. $f5 = \{a, \{b, c\}, \{d, \{e, f\}\}\}$. The Fig. 2g represents a framel $f6 = \{\{a, b\}, \{b, c\}\}$ where its two elements share the object b. Finally, the Fig. 2h represents the framel $f7 = \{\{b, c\}, \{d, \{e, f\}\}, \{c, \{e, f\}\}\}$, where the last two elements share the framel c while the first and the third elements share the framel $\{e, f\}$.

A main property defined on a framel is that of *membership*. The members of a framel f are all the framels that compose it, at each level of nesting. Then, we define the *memberset* of a framel as the set of all its members. As a particular case, we assume that an object has itself as its unique member.

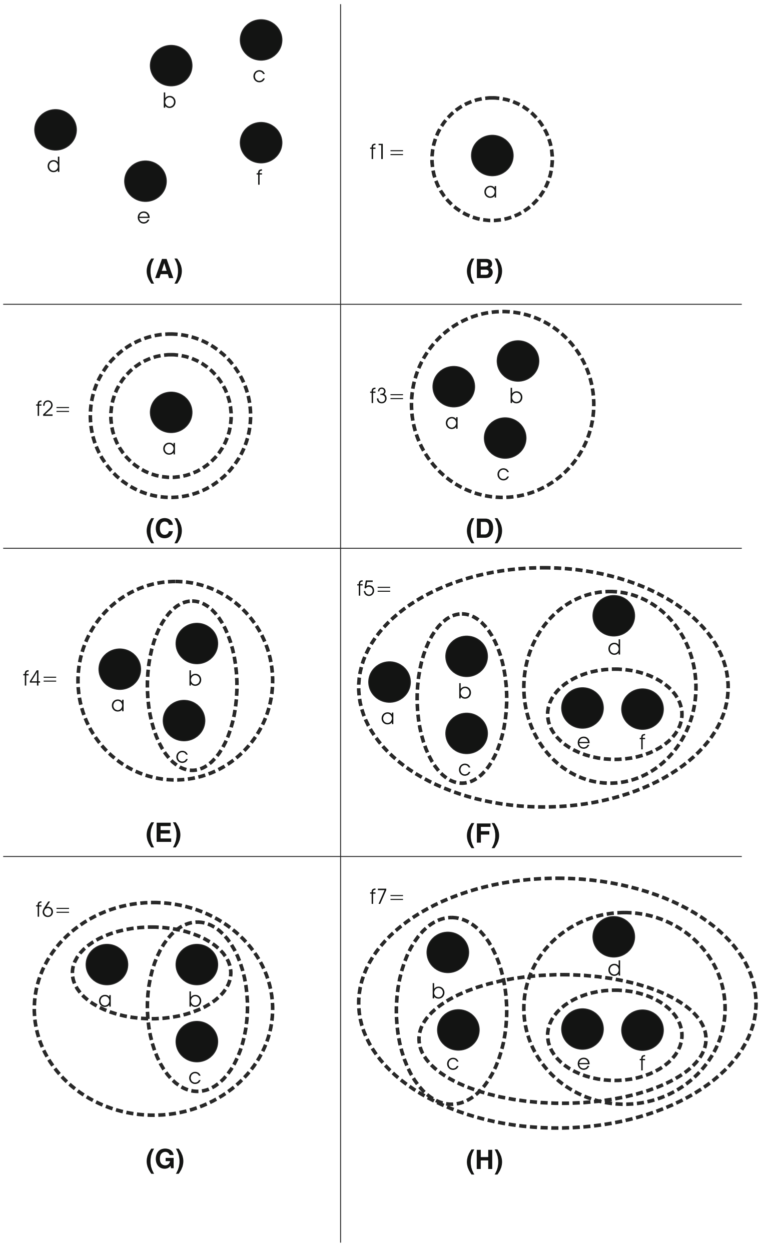


Fig. 2. a A set O of objects; **b–h** some example of framels on O

Definition 3.3 (*Memberset*) Let O be a set of objects and let f be a framel on O. The *memberset* of f, denoted by $\mathcal{M}_f$, is a set of framels on O where $\forall\, g \in \mathcal{M}_f$ either (i) $g \in f$ or (ii) $\exists\, k \in f$ such that $k \in \mathcal{M}_k$. If $f \in O$, then $\mathcal{M}_f = f$.

Example 3.4 (Membersets) The membersets of the framels depicted in Fig. 2b–f are:
Figure 2b: $\mathcal{M}_{f1} = \{a\}$.
Figure 2c: $\mathcal{M}_{f2} = \{a, \{a\}\}$.
Figure 2d: $\mathcal{M}_{f3} = \{a, b, c\}$.
Figure 2e: $\mathcal{M}_{f4} = \{a, b, c, \{b, c\}\}$.
Figure 2f: $\mathcal{M}_{f5} = \{a, b, c, d, e, f, \{b, c\}, \{e, f\}, \{d, \{e, f\}\}\}$.

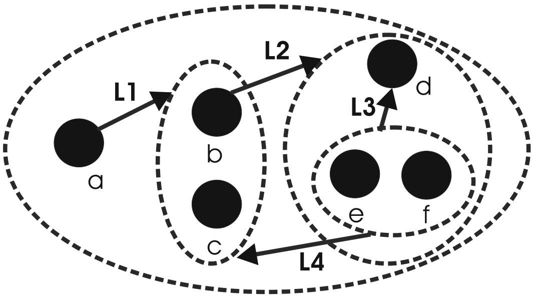


Fig. 3. An example of framoid

Figure 2g: $\mathcal{M}_{f6} = \{a, b, c, \{a, b\}, \{b, c\}\}$.
Figure 2h: $\mathcal{M}_{f7} = \{b, c, d, e, f, \{b, c\}, \{e, f\}, \{d, \{e, f\}\}, \{c, \{e, f\}\}\}$.

Property 3.5 (Member of level l) *Let O be a set of objects and f be a framel on O. We say that $g \in \mathcal{M}_f$ is a* member of level 0 *of f if $g \in f$. We say that g is a* member of level l *of f if $\exists\, k \in \mathcal{M}_f$ such that both $g \in k$ and k is a member of level $l - 1$ of f.*

Example 3.6 (Members) Consider the example of Fig. 2. The framel a is a member of level 0 of $f1$, the framel $\{a\}$ is a member of level 0 of $f2$, the framels a, b, c are members of level 0 of $f3$, the framels a and $\{b, c\}$ are framels of level 0 of $f4$, the framels a, $\{b, c\}$, $\{d, \{e, f\}\}$ are members of level 0 of $f5$, the framels $\{a, b\}$ and $\{b, c\}$ are framels of level 0 of $f6$ and the framels $\{b, c\}$, $\{c, \{e, f\}$, $\{d, \{e, f\}\}\}$ are framels of level 0 of $f7$. Analogously, a is a member of level 1 of $f2$, b and c are members of level 1 of $f4$, b, c, d, $\{e, f\}$ are members of level 1 of $f5$ and $f7$, a, b and c are members of level 1 of $f6$. Finally, e and f are members of level 2 of both $f5$ and $f7$.

Based on the notion of framel, we now define the notion of *framoid*. A framoid consists of a framel and a set of labelled arcs that connects some of the members of the framel. The structure of a framoid thus appears as an extension of a direct labelled graph, with the difference that the "nodes" of a framoid are the members of its framel, that instead of necessarily representing a single object can have a more complex structure, with possible levels of nesting.

Definition 3.7 (*Framoid*) A *framoid* is a triple $\langle O, f, A\rangle$, where O is a set of objects, f is a framel on O and A is a set of labelled arcs, such that each $a \in A$ is an ordered triple $\langle x, y, l\rangle$, where $x, y \in \mathcal{M}_f$, and l is a label. We denote by ø the cardinality of O, by n the cardinality of $\mathcal{M}_f$ and by α the cardinality of A.

Example 3.8 (Framoid) The framoid of Fig. 3 is equal to $\langle O, f5, A\rangle$, where $O = \{a, b, c, d, e\}$, $f5$ is the framel graphically depicted in Fig. 2f and $A = \{e1, e2, e3, e4\}$, such that $e1 = \langle a, \{b, c\}, L1\rangle$, $e2 = \langle b, \{d, \{e, f\}\}, L2\rangle$, $e3 = \langle\{e, f\}, d, L3\rangle$, and $e4 = \langle\{e, f\}, \{b, c\}, L4\rangle$.

We define two types of relationships on a framoid, called *membership* and *link*. A *membership* in a framoid is a relationship between two framels a and b members of f, such that b is member of a. A *link* in a framoid is a relationship between two framels a and b members of f, such that there exists an arc oriented from a to b.

Definition 3.9 (*Memberships*) Let $F = \langle O, f, A\rangle$ be a framoid. The *memberships* of F, denoted by $memberships_F$ is a relationship on $\mathcal{M}_f \times \mathcal{M}_f$ that contains all the ordered pairs $\langle a, b\rangle$, where $a, b \in \mathcal{M}_f$ and $b \in \mathcal{M}_a$.

Definition 3.10 (*Links*) Let $F = \langle O, f, A\rangle$ be a framoid. The *links* of F, denoted by $links_F$ is a relationship on $\mathcal{M}_f \times \mathcal{M}_f$ that contains all the ordered triple (a, b, l), where $a, b \in \mathcal{M}_f$ and there exists an arc $a = \langle a, b, l\rangle \in A$.

Note that between two framels a and b, members of a framoid F, only an instance (a, b) can exist in $memberships_F$, while many instances $\langle a, b, l\rangle$ can exist in $links_F$.

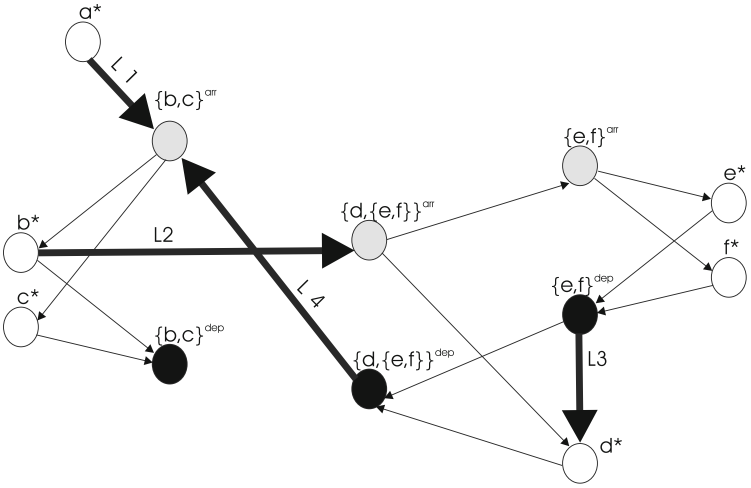


Fig. 4. The graph representation of the framoid depicted in Fig. 3

A framoid has got a number of members equal to the cardinality of $\mathcal{M}_f$, i.e. n. In its turn, each of these member, say m, has got a number of members equal to $|\mathcal{M}_m|$. Therefore, the cardinality of $memberships_F$, that we denote by ψ, is equal to

$$\psi = \sum_{m \in f} |\mathcal{M}_m| \tag{1}$$

It is possible to provide a representation of a framoid $F = \langle O, f, A\rangle$ by using a labelled directed graph that contains, for each object o of F a correspondent node $o*$, and for each member m of f two nodes m^{arr} and m^{dep}, called *arrival node* and departure node, respectively, where if m is an object then $m^{arr} = m^{dep} = m*$. For each arc a of A, oriented from a framel x to a framel y, a corresponding arc, called *link arc*, directed from x^{dep} to y^{arr} and labelled with the same label of a, is inserted in the graph-representation. Moreover, each node m^{arr} is linked by a fictitious arc, called *membership arc*, with the arrival node of each element of m, to represent the fact that each arc incoming in m^{arr} has to be joined with each element of m. Analogously, each departure node of the framels contained in m is linked by another membership arc with the node m^{dep} to represent the fact that each element of m is joined with each arc outcoming from m^{dep}. A conventional label $MEMBER$ is applied to all the membership arcs.

Definition 3.11 (*Graph-representation*) Let $F = \langle O, f, A\rangle$ be a framoid. The graph-representation of F, denoted by G_F, is the labelled directed graph $\langle N_F, A_F\rangle$, where (i) for each object $o \in O$ a correspondent node $o*$ is inserted in N_F and for each framel $m \in \mathcal{M}_f$, two nodes m^{arr} and m^{dep} are inserted in N_F such that $m^{arr} = m^{dep} = m*$ if $m \in O$ and (ii) for each pair of framels $x, y \in \mathcal{M}_f$ such that $y \in \mathcal{M}_x$, both an arc $\langle x^{arr}, y^{arr}, MEMBER\rangle$ and an arc $\langle y^{dep}, x^{dep}, MEMBER\rangle$ are inserted in A_F (these two arcs are called *membership arcs* and (iii) for each arc $\langle x, y, l\rangle \in A$, an arc $\langle x^{dep}, y^{arr}, l\rangle$ is inserted in A_F.

Example 3.12 (Graph-representation) Figure 4 shows the graph-representation of the framoid depicted in Fig. 3. The white circles represent nodes associated to objects, while grey (resp. black) circles represent arrival (resp. departure) nodes. Finally, the thin lines represent membership arcs while the bold lines represent link arcs.

It is simply to prove the following proposition:

Proposition 3.13 (Number of arcs in the graph-representation) *Let $F = \langle O, f, A\rangle$ be a framoid. The number of arcs in the graph-representation G_F is equal to $2 \cdot \psi + \alpha$.*

Proof. Directly derives from the consideration that for each membership of F two membership arcs are inserted in A_F, and for each arc in A an arc is inserted in A_F. □

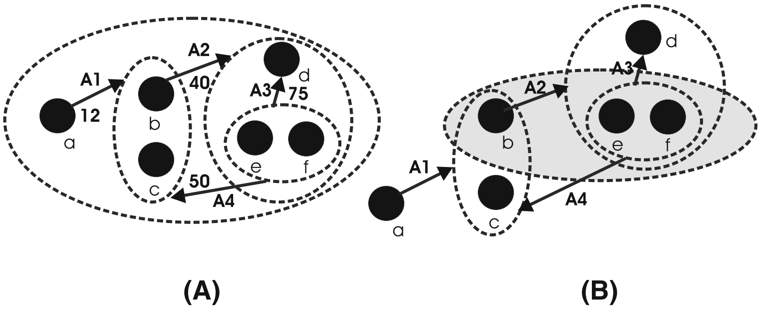


Fig. 5. a A framoid and **b** its unique strongly connected component (in grey)

It is possible to define for a framoid the notion of *path* between two framels.

Definition 3.14 (*Path*) Let $F = \langle O, f, A\rangle$ be a framoid and x, y be two members of f. We suppose to have two functions, namely $ini : A \rightarrow \mathcal{M}_f$ and $fin : A \rightarrow \mathcal{M}_f$ such that for each arc $e = \langle x, y, l\rangle \in A$, we have $ini(e) = x$ and $fin(e) = y$. A *path* between x and y is a sequence of arcs $a_1, a_2, \ldots, a_k \in A$, such that $x \in \mathcal{M}_{ini(a_1)}$, $y \in \mathcal{M}_{fin(a_k)}$ and $ini(a_{i+1}) \in \mathcal{M}_{fin(a_i)}, \forall i = 1, 2, k-1$.

Since the arcs of a framoid are labelled, and the label of an arc represents an information characterizing the relationship between the framels linked by the arc, we introduce the notion of $\mathcal{F}$-*relevant path*, that is a path whose arcs present values of the labels satisfying a given boolean function $\mathcal{F}$.

Property 3.15 ($\mathcal{F}$-relevance of a path) *Let $F = \langle O, f, A\rangle$ be a framoid, $p = a_1, a_2, \ldots, a_k \in A$ be a path in F and $\mathcal{F}$ be a boolean function that accepts as input a path and returns either* true *or* false. *The path p is called* $\mathcal{F}$-relevant *iff $\mathcal{F}(p) = true$.*

We can also define other properties for a framoid, analogously to similar properties of a graph, as that of *connection* between two framels and that of *strongly connected component*.

Property 3.16 (Connection) *Let $F = \langle O, f, A\rangle$ be a framoid and $x, y \in \mathcal{M}_f$ be two members of f. Moreover, let $\mathcal{F}$ be a boolean function accepting as input a path of F. We say that x and y are* connected *(resp.* $\mathcal{F}$-relevant connected*) if there exists at least a path (resp. a* $\mathcal{F}$-relevant path*) in F from x to y. Each member $x \in \mathcal{M}_f$ is connected (resp.* $\mathcal{F}$-relevant connected*) with itself.*

Example 3.17 (Path) Consider the framoid of Fig. 3. An example of path from a to c is $p = e1, e2, e4$ where $e1$ (resp. $e2$, $e4$) is the arc labelled as $L1$ (resp. $L2$, $L4$). In fact, note that the source node a is the initial node of the arc $e1$ and b, that is the initial node of $e2$, is in the memberset of the framel $\{b, c\}$, that is the final node of $e1$, while $\{e$,$f\}$, that is the initial node of $e4$, is in the memberset of {d,{e,f}}, that is the final node of $e2$ and finally c, that is the destination node of the path is in the memberset of $\{e$,$f\}$, that is the final node of $e4$.

Definition 3.18 (*Strongly* ($\mathcal{F}$-relevant) *connected components*) Let $F = \langle O, f, A\rangle$ be a framoid and let $\mathcal{F}$ be a boolean function accepting as input a path of F. A strongly connected component (resp. a strongly $\mathcal{F}$-*relevant connected component*) of F is a framel f^* on O^* where (i) $O^* \subseteq O$ and (ii) $m \in \mathcal{M}_f \; \forall \, m \in \mathcal{M}_{f^*}$ and (iii) for each oriented pair of framels (a, b), where $a, b \in f^*$, we have that a and b are connected (resp. F-*relevant connected*) in F.

The computational cost to find the strongly connected components of a framoid depends on both the number of memberships and the number of links present in it.

Theorem 3.19 (Finding strongly connected components) *Let $F = \langle O, f, A\rangle$ be a framoid. The time computational complexity to find the strongly connected components of F is $\mathcal{O}(\psi + \alpha)$.*

Proof. It is sufficient to consider that the time computational complexity of finding the strongly connected components of a directed graph having m arcs is $\mathcal{O}(m)$ and the problem of finding the strongly connected components of F is equivalent to that of finding the strongly connected components of the graph-representation of F, that has a number of arcs equal to $2 \cdot \psi + \alpha$ (see Proposition 3.13). □

Corollary 3.20 (Finding relevant strongly connected components). *Let $F = \langle O, f, A\rangle$ be a framoid, and $\mathcal{F}$ be a boolean function accepting as input a path in F. The time computational complexity to find the $\mathcal{F}$-relevant strongly connected components of F is $\mathcal{O}(\psi + \alpha)$.*

Proof. It directly derives from Theorem 3.19, and from the consideration that, in order to verify that each path p determined during the search of the strongly connected components is such that $\mathcal{F}(p) = true$, it is necessary to consider up to α arcs. □

Example 3.21 ($\mathcal{F}$-relevant strongly connected components). Consider the framoid of Fig. 5a, and the boolean function $\mathcal{F}(p)$ accepting as input a path p and returning $true$ if all the arcs composing p have a label value greater than 30. It is easy to see that its unique $\mathcal{F}$-relevant strongly connected component is the framel $\{b, \{e, f\}\}$, highlighted by a grey ellipse in Fig. 5b, since b is connected to $\{e, f\}$ being linked by the arc $A2$ (having label value equal to 40) to $\{d, \{e, f\}\}$ and $\{e, f\}$ is connected to b being linked by the arc $A4$ (having label value equal to 50) to $\{b, c\}$. For better understanding this result, it is sufficient to apply the standard algorithm for finding the strong connected components to the graph representation of this framoid, depicted in Fig. 4.

4. Discovering naturally emerging framels in a framoid

It is interesting to point out that the framel $\{b, \{e, f\}\}$ determined as the unique $\mathcal{F}$-relevant strongly connected component in the framoid of Fig. 5a is not a member of the framoid. In other words, determining such a framel as the result of finding the $\mathcal{F}$-relevant strongly connected components of the framoid has led us to discover a structure embedded in the framoid, not explicitly "declared" as a part of the framoid, and that naturally emerges in consequence of the arc-relationships and member-relationships existing in the framoid itself. We call such a type of framel a *naturally emerging framel* (NE-framels, for short).

Definition 4.1 (*Naturally emerging framels*) Let $F = \langle O, f, A\rangle$ be a framoid and $\mathcal{F}$ be a boolean function accepting as input a path of F. A *naturally emerging framel* (NE-framel) on $\mathcal{F}$ of F is a $\mathcal{F}$-relevant strong connected component c of F such that $c \notin \mathcal{M}_f$.

Theorem 4.2 (Finding the NE-framels) *Let $F = \langle O, f, A\rangle$ be a framoid and $\mathcal{F}$ be a boolean function accepting as input a path of F. The time computational complexity to find the NE-framels on $\mathcal{F}$ of F is $\mathcal{O}(n^2)$.*

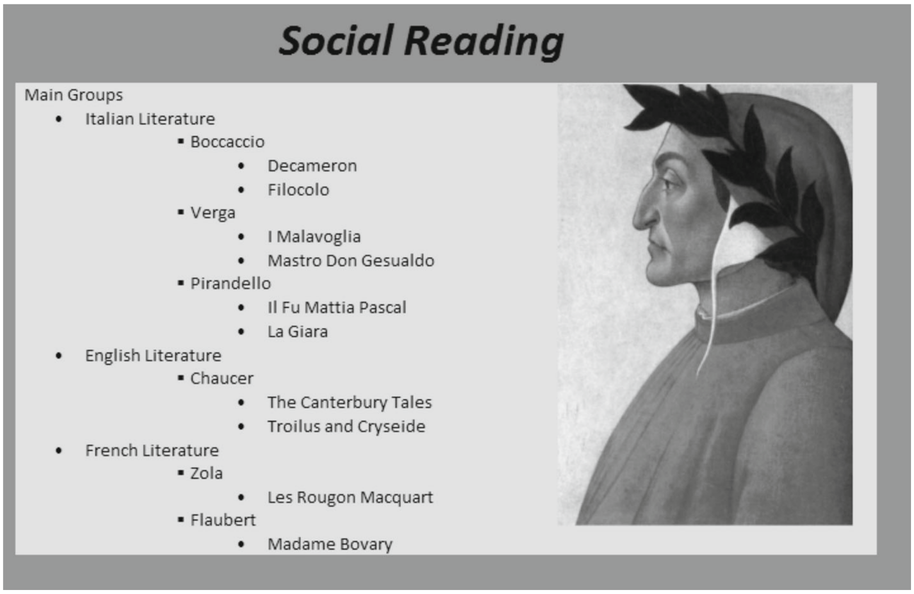
Proof. The time computational complexity of finding the NE-framels of F derives from two contributions, namely (i) that of finding the $\mathcal{F}$-relevant strongly connected components of F and (ii) that of checking, for each of these components, if it belongs to $\mathcal{M}_f$.

Regarding the contribution (i), the time computational complexity of finding the $\mathcal{F}$-relevant strongly connected components of F is $\mathcal{O}(\psi + \alpha)$ (see Corollary 3.20). We observe that both ψ and α are $\mathcal{O}(n^2)$, therefore the computational complexity of this task is $\mathcal{O}(n^2)$.

Regarding the contribution (ii), observe that the task of checking if a $\mathcal{F}$-relevant strongly connected component c belongs to $\mathcal{M}_f$ is $\mathcal{O}(n)$ (where n is the cardinality of $\mathcal{M}_f$), since it is necessary to check if there exists an element e of $\mathcal{M}_f$, such that both $\mid c \mid = \mid e \mid$ and all the elements of c belong to e. Considering that we have a number of $\mathcal{F}$-relevant strongly connected components of F that is $\mathcal{O}(n)$, the overall time computational complexity for executing the checking above for all the $\mathcal{F}$-relevant strongly connected components is $\mathcal{O}(n)^2$. □

The theorem above shows that the unique variable that influences the computational complexity of finding the NE-framels of a framoid is the value n, which intuitively represents the number of distinct groups existing in the framoid, regardless the values α and ψ, that instead represents the number of relationships (links and memberships, respectively) present in the framoid.

Finding naturally emerging framels in a framoid F allows us to discover new, potentially interesting information about the framoid from a semantic viewpoint. Indeed, if we assume that all the elements of a framel are semantically related, a naturally emerging framel f determined into F represents a semantic relationship between its elements, and thus it can be considered as a new "semantic frame" individuated in the environment represented by the framoid F. We will see, in the next section, some examples of how such a kind of information can be usefully exploited in the context of the Semantic Web.

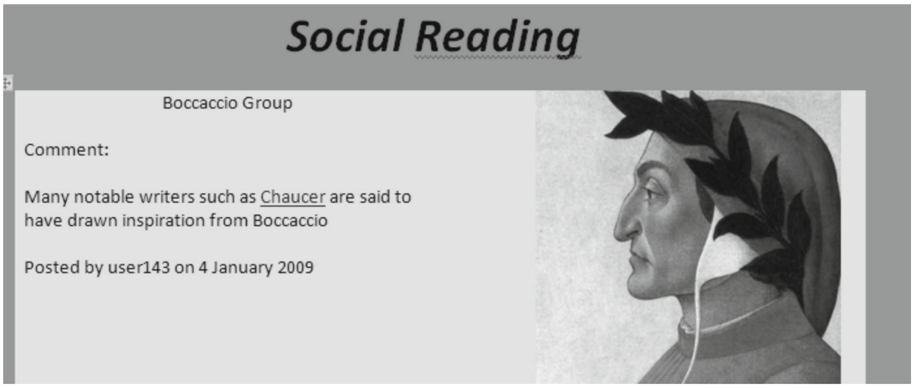


Social Reading

Main Groups

- Italian Literature
  - Boccaccio
    - Decameron
    - Filocolo
  - Verga
    - I Malavoglia
    - Mastro Don Gesualdo
  - Pirandello
    - Il Fu Mattia Pascal
    - La Giara
- English Literature
  - Chaucer
    - The Canterbury Tales
    - Troilus and Cryseide
- French Literature
  - Zola
    - Les Rougon Macquart
  - Flaubert
    - Madame Bovary

Fig. 6. A social network focused on books



Social Reading

Boccaccio Group

Comment:

Many notable writers such as Chaucer are said to have drawn inspiration from Boccaccio

Posted by user143 on 4 January 2009

Fig. 7. A link to group "Chaucer" in a comment of the group "Boccaccio"

5. Using framoid representation in the Semantic Web: recommendations in social networks

In this section, we describe how it is possible to use framoids to represent a well-known reality in the Semantic Web, that is the generation of recommendations for the users of a social network.

A social network is a virtual environment where each member can contact other members in a context characterized by a high social acquaintance. Generally, the members of a social network can discuss on particular topics, and often there are groups of discussion for each topic, that can be hierarchically organized. As an example, suppose that in a social network focused on books and literature (see Fig. 6), there are three main groups called "Italian", "English" and "french", dealing with Italian, English and french literature, respectively. Moreover, suppose that the group "Italian" is partitioned in three subgroups called "Boccaccio", "Verga" and "Pirandello", focused on the homonymous Italian writers, while the group "English" is composed only by the group "Chaucer", that deals with the English writer and, finally, the group "french" contains the three subgroups "Zola", "Balzac" and "Flaubert", associated to the homonymous french writers. Furthermore, suppose that each subgroup associated to a writer, for example "Chaucer", is further partitioned in sub-subgroups associated to some works of that writer; for example, in the case of Chaucer, we have a sub-subgroup for discussing about "The Canterbury Tales" and another sub-subgroup focused on "Troilus and Cryseide".

We suppose that a Web page is associated with each group, subgroup or sub-subgroup, where users can publish textual comments, multimedia documents and so on. It is also possible to insert in a textual comment a link to another group, as in Fig. 7 where a comment in the subgroup “Boccaccio” contains a link to the group “Chaucer”.

It is possible to provide a user of the social network with a set of recommendations regarding the most suitable pages to visit. Generally, the recommender systems proposed in the literature generate these recommendations using a content-based and/or a collaborative filtering algorithm. To this purpose, several approaches as that presented in [RoS06] associates a user’s profile to each user, containing a set of pairs $\langle g, i_g \rangle$, where g is the name of a group/subgroup/sub-subgroup and i_g is a value representing the user’s interest for g, such that the higher is the value of i_g, the higher is the user’s interest in g. For instance, we can assume that $i_g = 1$ means minimum interest in g, while $g_i = 5$ means maximum interest in g and values between 1 and 5 mean intermediate degrees of interest. A recommender system can use this user’s profile to generate recommendations for u following two main approaches:

Content-based approach A recommender system using a content-based approach can suggest to the user the pages belonging to those groups g in which he is most interested, accordingly to the values of i_g. As an example, suppose that a user u a profile $P_u = \left\{ \langle \text{“}Boccaccio'', 5 \rangle \, \langle \text{“}\,Verga'', 4 \rangle, \langle \text{“}Pirandello'', 1 \rangle \right\}$. Then, suppose that the recommender system is set to recommend pages of those groups in which the interest is greater than 3. Consequently, the recommender system will suggest to u the page “The Canterbury Tales”, which is the only page contained in the group “Boccaccio”, and the pages “I Malavoglia” and “Mastro Don Gesualdo”, belonging to the group “Verga”.

Collaborative filtering approach The recommender systems can also use a collaborative filtering algorithm, for suggesting to the user u those pages that he did not access in the past and that instead are accessed by other users having a profile similar to that of u. For instance, suppose that in the social network of Fig. 6 there is a user x having the following profile: $P_x = \left\{ \langle \text{“}Boccaccio'', 5 \rangle, \langle \text{“}\,Verga'', 3 \rangle, \langle \text{“}Pirandello'', 2 \rangle, \langle \text{“}Beaudelaire'', 5 \rangle \right\}$. This profile can be compared with that of u in order to discover a possible similarity between u and x. A widely used similarity measure is the Jaccard’s measure, defined as the ratio between the number of items shared by the two profiles and the total number of distinct items. In the case of u and x, the two profiles share the three items “Boccaccio”, “Verga” and“Pirandello”, while the number of distinct items is 4 (i.e., “Boccaccio”, “Verga”, “Pirandello” and “Beaudelaire”), then the Jaccard’s measure is equal to 0.75. Supposing that in our case the collaborative filtering algorithm considers as “similar” to u only those users having a Jaccard’s measure equal to 0.7, then x will be considered as a user similar to u. Moreover, we suppose that the algorithm recommends the two pages most accessed by x and that in our case these pages are “Filocolo” and “Les Fleurs du Mal”. Consequently, the algorithm will recommend to u these two pages.

We can easily see the limitations of both the two approaches above. The content-based algorithm suggests the user those pages belonging to groups of his interest, but it is unable to discover novel groups that might be potentially interesting for the user, that simply did not accessed them in the past and consequently does not know them. More in particular, the content-based approach does not exploit semantic relationships possibly existing between groups. The collaborative filtering approach is able to discover new, potentially useful, information coming by users having interests similar to those of u. However, this information is purely based on the accesses performed by these similar users, and do not take into account any semantic relationships between pages. In the example above, the collaborative filtering algorithm suggests to u the pages “Filocolo” and “Les Fleurs du Mal” based on the fact the similar user x accessed them in the past, but any search of semantic closeness between these pages and the u’s interests has been performed.

However, in a context as that of social networks, semantic relationships between pages often exist and should be exploited. In particular, links between pages can be considered as useful information about semantic relationships. Another useful information can be represented by the percentage of users that select a link in a page. Modeling the social network as a framoid can make possible to capture these relationships. In particular, we propose to model each page by an object, and each group by a framel (that can be, as a particular case, an object). We consider that each framel is associated to a representative Web page, and the actual links present in the Web pages are modelled by arcs in the framoid. The label of an arc represents the percentage of selections of the associated link performed by the users visiting the Web page. Moreover, we model the isa-relationship between a sub-group b and its super-group a by including the framel associated to b in the framel associated to a. For instance, in Fig. 8a, it is shown the framoid representation of the social network depicted in Fig. 6.

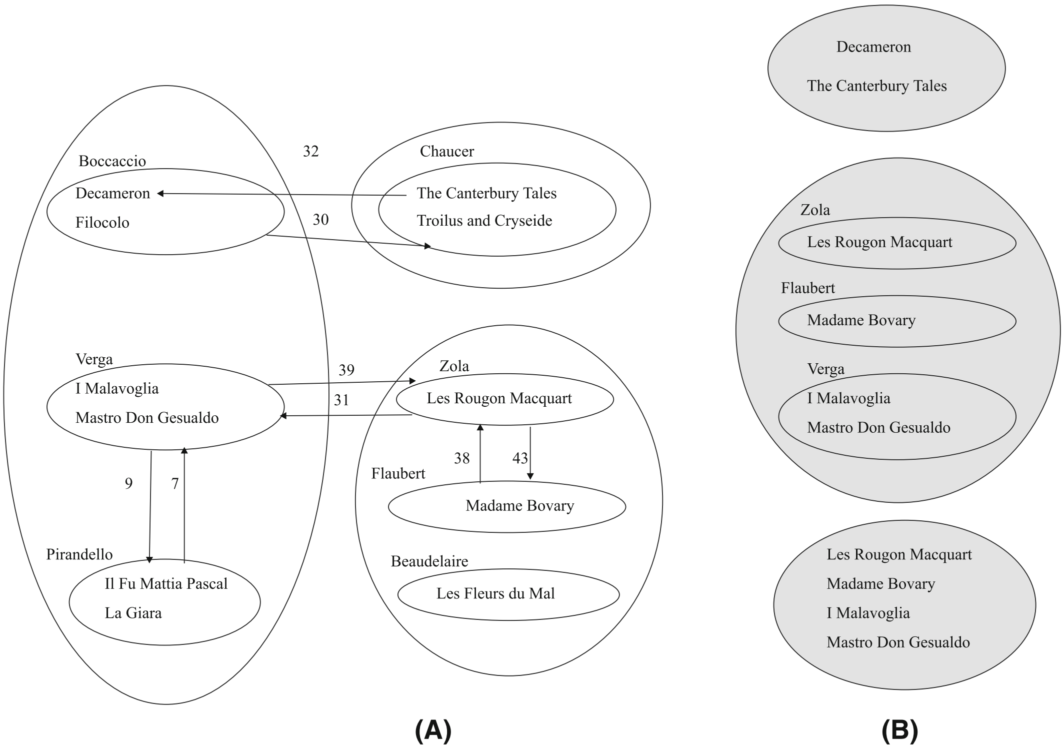


Fig. 8. a A framoid associated to a literature social network; **b** Three naturally emerging framels

Now, in order to exploit such a representation to generate recommendations for the user u taking into account semantic relationships between the Web pages, we could compute the NE-framels on $\mathcal{F}$ of the framoid, where $\mathcal{F}$ is the boolean function that accepts as input a path and returns $true$ if all the labels of the path have a value greater than 30. This means to consider two pages as semantically connected only if each link composing the path between them has been selected by the users at least in the 30 percent of the cases. The resulting NE-framels are represented in Fig. 8b. These framels represent novel, potentially useful, information about semantic closeness of Web pages. For instance, we discover that the pages “Decameron” and “The Canterbury Tales” are semantically related, as well as the pages “I Malavoglia”, “Mastro Don Gesualdo”, “Les Rougon-Macquart” and “Madame Bovary”. Then, supposing that u accessed in the past “Decameron”, it is possible to suggest it “The Canterbury Tales” that is in one of the new framels discovered by our algorithm and that also contains “Decameron”. Similarly, if u also accessed in the past “I Malavoglia”, we can suggest him “Mastro Don Gesualdo”, “Les Rougon-Macquart” and “Madame Bovary”, that are contained together with “I Malavoglia” in another discovered framel. Finally, observing that a third discovered framel groups together “Verga”, “Zola” and “Flaubert”, we can suggest u, that is interested in “Verga”, to visit the pages representative of “Zola” and “Flaubert”.

Note that “Mastro Don Gesualdo” has been also suggested by the content-based algorithm, while the other recommendations we have generated based on semantic relationships were not suggested by the traditional approach. Interestingly enough, none of the collaborative filtering recommendations are suggested by our approach, since none of them contains semantic relationships with the pages accessed by u. This leads to argue that traditional collaborative filtering can generate recommendations that are not semantically related in any way with the interests of the involved user. However, this does not mean that the recommendations generated by the collaborative filtering approach are necessarily ineffective, but only that our approach can be considered as a possible integration of the collaborative filtering based, on semantic considerations.

6. Evaluation

In this section we describe some experiments we have performed in order to evaluate the advantages introduced by our approach in the detection of semantic associations. We have chosen to apply our method in the field of recommender systems to support users' Web navigation, that appears one of the most suitable for exploiting semantic associations in presence of groups of entities.

As described in Sect. 5, most of the existing recommender systems generate both content-based and collaborative filtering recommendations. In our experiment, we have considered the recommender system MUADDIB [RSG09, Mua09] (formerly MASHA [RoS06]), that generates highly effective recommendations.

We have added to MUADDIB the capability to generate semantic associations-based recommendations exploiting the framoid-based approach that we propose in this paper. Moreover, we have also implemented on MUADDIB the generation of recommendations based on traditional semantic associations, exploiting RDF graphs, and we have compared these two different approaches for evaluating our contribution.

MUADDIB is an agent-based virtual community. It contains a set of registered users, and a set of registered Web pages. Each page is an XML document, and the page elements are instances of common *topics* contained in a central XML-Schema Ontology. In particular, in our experiments we have used the publicly available Italian-English Literature Dictionary [Mua09], that contains a set of 94 different topics related to authors of Italian and English literature. Moreover, we have exploited a set of 300 different XML Web sites which are based on the aforementioned dictionary (also these sites are downloadable at the MUADDIB site). This way it is possible, for a software agent, to understand if a page visited by a user deals with Shakespeare, or Keats, or whatever other topic contained in the common ontology. Each user u joined with the community is monitored by a personal agent, that build a *profile* of u, denoted by P_u, containing the interests of u. Roughly speaking, the profile P_u is a list of all the *topics* which u is interested in. Moreover, for each of these interesting topics, say t, a *coefficient of interest* c_t ranging in the real interval [0,1], is associated to represent how much the user is interested in that topic. Therefore, the profile P_u is a list of pairs topic-interest coefficient $\langle t, c_t \rangle$. The recommendations for each user u are computed by a recommendation algorithm, that performs the following activities: (i) comparing the profile P_u with the pages contained in the site, suggesting to the user those pages whose topic best match with his interests; (ii) comparing the profile P_u with the profiles of the other users present in the community, suggesting to u those pages mostly accessed by those users whose profiles best match with P_u. For all the details of the recommendation algorithm, see [RoS06].

As navigational data, we have used the datasets *training* and *test* available at the MUADDIB site. These datasets contains the logs of 200 real (distinct) users, denoted by $u_1, u_2, \ldots, u_{200}$. To study how the performances depend on the number of monitored users, the users have been partitioned on different sub-sets called S1, S2, S3 and S4, where the set S1 contains the first 50 users, i.e. $S1 = \{u_1, u_2, \ldots, u_{50}\}$, the set S2 contains the first 100 users, i.e. $S2 = \{u_1, u_2, \ldots, u_{100}\}$, the set S3 contains the first 150 users, i.e. $S3 = \{u_1, u_2, \ldots, u_{150}\}$ and finally the set S4 contains all the 200 users, i.e. $S4 = \{u_1, u_2, \ldots, u_{200}\}$. For each user, the dataset *training* stores the first 900 accessed URLs, in order to construct the user's profile, while the dataset *test* contains other 600 to be used in the test phase. Each user's access in the datasets has been represented by a tuple $\langle u, t, \tau, d \rangle$ where u is the identifier of the user, t is the topic associated with the accessed URL, τ is the no-idle time spent on the page associated with the accessed URL and d is the exploited device. These information are exploited by the MUADDIB recommendation algorithm to compute the interest coefficient of each topic.

6.1. Training phase

The dataset *training* has been exploited in the training phase. In this phase, the MUADDIB agents have built the personal profiles of their associated users, in the way described in [RoS06]. Moreover, during this phase, the framoid F associated to the virtual community has been built following the indications of an expert of the domain. This framoid represents additional information about semantic relationships existing between different topics. It contains 109 framels, such that 94 of them are singleton framels associated to the topics of the common ontology and other 15 framels are groups of semantically related topics. For instance, there is a framel called $f10$ that contains all the Italian XIX century authors, while another framel $f11$ contains the English XIX century authors and there is also a framel $f12$ that contains as members both $f10$ and $f11$. Moreover, the framoid contains 99 arcs, representing directed relationships between framels. An arc between two framels f_i and f_j has been added if there exists in the site set a Web page p_i associated with a topic belonging to f_i and having a hyperlink to a

Web page p_j associated with a topic belonging to f_j. During the training-phase, based on the analysis of the log file *training*, for each arc $a = \langle f_i, f_j, l_a \rangle$ of the framoid the label value l_a is computed, representing the usage level of the hyperlinks associated with a. More in particular, l_a is determined by considering the set U_a of the users that have selected, in the history of the virtual community represented in the file *training*, at least one time a hyperlink connecting two Web pages p_i and p_j, such that p_i (resp. p_j) is associated with a topic belonging to f_i (resp. p_j). For each of those users $u \in U_a$, is computed the percentage $\frac{n_a^u}{N_u}$, where n_u^a is the number of times u_a selected a hyperlink connecting two Web pages p_i and p_j as above, and N_u is the total number of hyperlink selection performed by u_a. Then, the average of all the contributions $\frac{n_a^u}{N_u}$ is computed and used as label for the arc a, that is:

$$l_a = \frac{\sum_{u \in U_a} \frac{n_a^u}{N_u}}{|U_a|} \tag{2}$$

Moreover, after having built F, we consider the set N of nodes representing the singleton framels of the framoid (i.e., the single objects) and the subset A of the arc set of the framoid such that each arc of A connects two singleton framels. The structure $G = \langle N, A \rangle$ represents a classical RDF graph, that considers only the relationships between the single objects without taking into account the relationships involving groups.

In our experiments, we have added to the MUADDIB recommendation algorithm the capability of generating semantic associations-based recommendations. Therefore, the algorithm will suggest to the user that is visiting a site a set of Web pages that is the union of the set of content-based recommendations, the set of the collaborative filtering recommendations and the set of semantic associations-based recommendations.

The semantic associations-based recommendations are computed as follows.

6.2. Test phase

The framoid F built in the previous phase and the RDF graph have been then used in a test phase, for computing two different types of semantic associations-based recommendations.

Preliminary, in such a phase, for each user u, in correspondence of each tuple $p = \langle u, t, \tau, d \rangle$ corresponding to a Web page belonging to the test-set and visited by u, we have generated with the MUADDIB algorithm the recommendations M_p^{MUAD}, consisting of a set of suggested Web pages, determined by using content-based and collaborative-filtering methodologies.

In addition, we have determined the set M_p^{SA}, containing all the pages associated with topics that can be reached by a path starting from p on the RDF graph.

Moreover, we have used the framoid to compute the NE-framels on $\mathcal{F}$ of the framoid, where $\mathcal{F}$ is the boolean function that accepts as input a path and returns $true$ if all the labels of the arcs of the path have a suitable value, representing a reasonable percentage of usage up to which the link associated with an arc can be considered as relevant. Based on some preliminary experiments conducted on real users, we have set to 30 that value, that means to consider relevant a path if all the links composing it have been selected by the user at least the 30 percent of the times. Then, we have determined the set M_p^{GSA}, containing all the pages associated to topics belonging to the same NE-framel of the topic t associated to the page p.

Finally, we have computed the set $M_p^{MUAD+SA} = M_p^{MUAD} \bigcup M_p^{SA}$, that contains all the distinct Web pages belonging either to M_p^{MUAD} or to M_p^{SA}. This latter set contains all the recommendations generated by the traditional MUADDIB algorithm, together with possible other recommendations deriving by the semantic associations determined exploiting the RDF graph. Analogously, we have computed the set $M_p^{MUAD+GSA} = M_p^{MUAD} \bigcup M_p^{GSA}$, containing all the distinct Web pages belonging either to M_p^{MUAD} or to M_p^{GSA}. Such a set contains all the recommendations generated by the traditional MUADDIB algorithm, together with possible other recommendations deriving by the semantic associations determined by our approach that exploits the NE-framels.

Finally, we have compared the effectiveness of recommending to the user u the set M_p^{MUAD} with that of recommending the sets $M_p^{MUAD+SA}$ and $M_p^{MUAD+GSA}$, respectively.

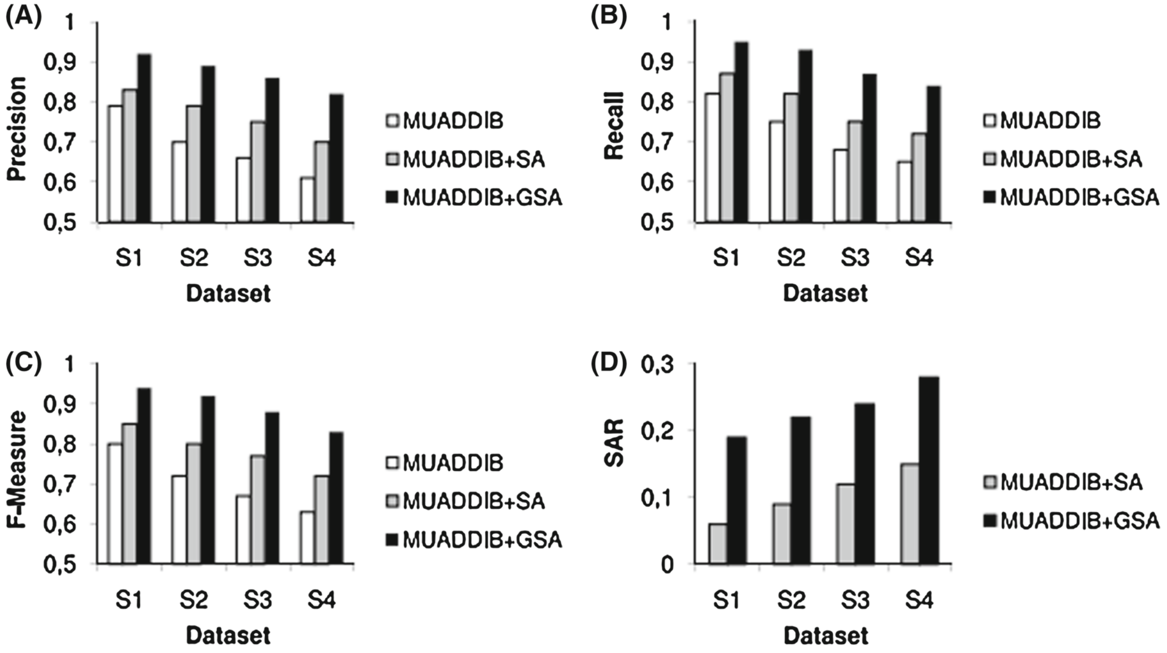


Fig. 9. [Comparison between MUADDIB, MUADDIB+SA and MUADDIB+GSA in terms of precision (**a**), recall (**b**) and F-measure (**c**). Percentage of semantic association-based recommendations of MUADDIB+SA and MUADDIB+GSA (**c**)

To this purpose, we have considered a recommended page r as *accepted* by the user u if it appears at least one time in the next 20 logs associated to u in the log file *test*. The value 20 has been chosen as a reasonable delay to consider the user's choice as actually influenced by the recommendation, on the basis of several apposite experiments we have conducted about the users' behaviour.

We have used, as a measure of effectiveness, the performance metrics, precision, recall and F-measure, accordingly with most of the related work [KAB04]. Precision is defined as the share of the pages actually accepted by u among those recommended by the recommendation algorithm; vice versa, Recall is the share of the pages suggested by the recommendation algorithm among those accepted by u. F-measure represents the harmonic mean between precision and recall. We call R_p the recommendations provided by a generic recommendation algorithm when the user visits the page p, and $next_p$ the 20 pages visited after p. Then Precision, recall, and F-measure can be represented as follows.

$$Pre(R_p) = \frac{|R_p \bigcap next_p|}{|R_p|} \tag{3}$$

$$Rec(R_p) = \frac{|R_p \bigcap next_p|}{|R_p|} \tag{4}$$

$$F(R_p) = \frac{2 * Rec(R_p) * Pre(R_p))}{Rec(R_p) + Pre(R_p))} \tag{5}$$

The histograms of Fig. 9 show the values of the average precision, the average recall and the average F-measure obtained, in this experiment, by the three considered approaches, for each of the four dataset $S1$, $S2$, $S3$ and $S4$. The histogram "MUADDIB" is associated to the use of the set $R_p = M_p^{MUAD}$ in the formula above, that is to use the traditional MUADDIB algorithm, while the histogram "MUADDIB with SA" corresponds to use $R_p = M_p^{MUAD+SA}$, i.e. exploiting the traditional semantic associations derived by the RDF graph. Finally, the histogram "MUADDIB with GSA" is generated using $R_p = M_p^{MUAD+GSA}$, i.e. exploiting the group-based semantic associations derived by the framoid analysis. The average has been computed on all the pages present in the test database.

The results graphically depicted in Fig. 9 show that the exploitation of the semantic associations introduces an improvement of the effectiveness with respect to the traditional MUADDIB recommendations, both in terms of precision (Fig. 9a) and recall (Fig. 9b), and consequently also in terms of F-measure. However, while the improvement introduced by the SA recommendations is relatively small, ranging between 6 and 14 percent in terms of F-measure (Fig. 9c), the improvement induced by the GSA recommendation is significantly more relevant, quantified in terms of F-measure as ranging between 16 and 29 percent. The histograms show that the larger the users' data set is, the higher the improvement due to the usage of GSA recommendation is. This direct dependence of the effectiveness of our approach on the number of users is intuitively understandable considering that semantic associations are effective when the relationships between the entities, represented by hyperlinks between Web pages, are sufficiently accessed by the users, and this access increases when the number of users increases too.

The capability of our approach of finding relevant recommendations that are not produced by the traditional MUADDIB algorithm, and that are not even determined by using classical semantic associations, is clearly shown in Fig. 9d. In percentage, the SA recommendations that are not produced by MUADDIB range in 6–15 percent, while the GSA recommendations that MUADDIB does not yield range in 19–28 percent.

7. Conclusion

Although a significant effort has been made in the recent past to exploit metadata for improving the usability of the Web, however most of the proposed approaches only focus to improve querying on RDF knowledge bases. As a result, the main result produced by these approaches consists in finding relationships between entities, represented by paths, or subgraphs, in RDF graphs. These relationships, often called semantic associations, are certainly useful to discover semantic links between single entities, thus supporting a semantic analysis of the knowledge bases, but they do not consider the structure of the information present on the Web. Such a structure, far from being composed of single entities, in most cases can be viewed as a hierarchical organization, where entities are collected into logically homogeneous groups, each group possibly containing nesting sub-groups. We find that the complexity of this structure is a precious source of information to improve our possibility of finding semantic associations in Web data, that involve, besides of single entities, also groups of entities. Starting from this consideration, we have proposed to model the Web knowledge bases by an apposite data structure, called framoid, able of both suitably representing the hierarchical organization of groups on data and maintaining the capability to express semantic relationships between objects, traditionally own by the graph structures. The framoid representation allowed us to show that it is possible to express the problem of finding semantic associations in presence of groups with an approach analogous to that of finding the strongly connected components in a direct labelled graph. We have theoretically characterized the components on a framoid that represent semantically meaningful aggregation of objects having a hierarchical organization, that we have called NE-framels. In our vision, NE-framels represent a way of representing semantic associations in presence of groups of entities. As a case study for exploiting NE-framels, we have analyzed the generation of recommendations for Web users. In particular, we have implemented our approach on the top of an existing recommender system, showing that the use of the NE-framels significantly improves the efficiency of the recommendations with respect to both the traditional approaches that do not use semantic associations and the classical approach to generating semantic associations without considering the presence of hierarchically structured groups.

The experiments also show that the advantages introduced by the approach become very relevant when the size of the users' community is large enough, since the approach is based on the use on a sufficiently complex structure of the Web sources that arises only in presence of a massive access to the available data.

As for our future work, we are planning to extend the study of the properties of the framoid data model, on the one hand, and to better characterize from a theoretical viewpoint the advantages and the limitations of our approach to discover semantic associations, on the other hand. We suppose that is possible to determine some relationship between the structure of a framoid and the semantic significance of the contained NE-framels, and we think that such a relationship could be suitably characterized under statistical considerations. The study of this kind of relationship is the main issue of our ongoing research.

References

[AND08] Aleman-Meza B, Nagarajan M, Ding L, Sheth AP, Arpinar IP, Joshi A, Finin T (2008) Scalable semantic analytics on social networks for addressing the problem of conflict of interest detection. ACM Trans Web 2(1):1–29

[AMS05] Anyanwu K, Maduko A, Sheth A (2005) Semrank: ranking complex relationship search results on the semantic web. In: WWW '05: Proceedings of the 14th international conference on World Wide Web. ACM, New York, pp 117–127

[AnS02] Anyanwu K, Sheth A (2002) The ρ operator: discovering and ranking associations on the semantic Web. SIGMOD Rec 31(4):42–47

[AnS03] Anyanwu K, Sheth A (2003) P-queries: enabling querying for semantic associations on the semantic Web. In: WWW '03: Proceedings of the 12th international conference on World Wide Web. ACM, New York, pp 690–699

[Bar04] Barton S (2004) Designing indexing structure for discovering relationships in rdf graphs. In: Proceedings of the Dateso 2004 Annual International Workshop on DAtabases, TExts, Specifications and Objects, Desna, Czech Republic, April 14–16, 2004, volume 98 of CEUR Workshop Proceedings. CEUR-WS.org, pp 7–17.

[GaR08] Garruzzo S, Rosaci D (2008) Agent clustering based on semantic negotiation. ACM Trans Auton Adapt Syst 3(2):1–40

[GeD05] Getoor L, Diehl CP (2005) Link mining: a survey. SIGKDD Explor. Newsl 7(2):3–12

[GGE09] Groppe J, Groppe S, Ebers S, Linnemann V (2009) Efficient processing of sparql joins in memory by dynamically restricting triple patterns. In: SAC '09: proceedings of the 2009 ACM symposium on applied computing. ACM, New York, pp 1231–1238

[Hje01] Hjel J (2001) Creating the semantic web with RD. Wiley, New York

[KYK03] Kamei K, Yoshida S, Kuwabara K, Akahani J, Satoh T (2003) An agent framework for inter-personal information sharing with an rdf-based repository. In: Proc. of the Semantic Web—ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, volume 2870 of Lecture Notes in Computer Science. Springer, pp 438–452

[KAB04] Kim D, Atluri V, Bieber M, Adam N, Yesha Y (2004) A clickstream-based collaborative filtering personalization model: towards a better performance. In: Proceedings of the int. workshop on web information and data management. ACM, pp 88–95

[KoJ07] Kochut K, Janik M (2007) Sparqler: extended sparql for semantic association discovery. In: The semantic web: research and applications, 4th european semantic web conference, ESWC 2007, Innsbruck, Austria, volume 4519 of Lecture Notes in Computer Science. Springer, pp 145–159

[Mua09] MUADDIB Project URL (2009). http://www.muad.altervista.org.

[RMP05] Ramakrishnan C, Milnor WH, Perry M, Sheth AP (2005) Discovering informative connection subgraphs in multi-relational graphs. SIGKDD Explor. Newsl 7(2):56–63

[Rdf13] RDF W3C URL (2013). http://www.w3.org/rdf.

[RoS06] Rosaci D, Sarné GML (2006) Masha: a multi-agent system handling user and device adaptivity of web sites. User Model User Adapted Interact 16(5):435–462

[RSG09] Rosaci D, Sarné GML, Garruzzo S (2009) MUADDIB: a distributed recommender system supporting device adaptivity. ACM Trans Inf Syst 27(4):1–41

[ScS08] Schenk S, Staab S (2008) Networked graphs: a declarative mechanism for sparql rules, sparql views and rdf data integration on the web. In: WWW '08: Proceeding of the 17th international conference on World Wide Web. ACM, pp 585–594

[Sem09] Semantic Web W3C URL (2009). http://www.w3.org/2001/2w.

[SSB08] Stocker M, Seaborne A, Bernstein A, Kiefer C, Reynolds D (2008) Sparql basic graph pattern optimization using selectivity estimation. In: WWW '08: Proceeding of the 17th international conference on World Wide Web. ACM, New York, pp 595–604

[TTK08] Theoharis Y, Tzitzikas Y, Kotzinos D, Christophides V (2008) On graph features of semantic web schemas. IEEE Trans Knowl Data Eng 20(5):692–702

[ToF06] Tong H, Faloutsos C (2006) Center-piece subgraphs: problem definition and fast solutions. In: KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, pp 404–413

[WBB08] Wang W, Barnaghi PM, Bargiela A (2008) Search with meanings: an overview of semantic search systems. Int J Commun SIWN 3:76–82

[Wei09] Weikum G (2009) Harvesting, searching, and ranking knowledge on the web: invited talk. In: WSDM '09: proceedings of the second acm international conference on web search and data mining. ACM, New York, pp 3–4

[Zhu09] Zhuge H (2009) Communities and emerging semantics in semantic link network: discovery and learning. IEEE Trans Knowl Data Eng 21(6):785–799

Received 3 January 2013
Revised 31 March 2015
Accepted 31 May 2015 by Jim Woodcock
Published online 9 July 2015