AuthorRank+FOAF: Ranking for Co-Authorship Networks on the Web

Lule Ahmedi
Department of Computer Engineering
University of Prishtina
Prishtinë, Republic of Kosovo
lule.ahmedi@uni-pr.edu

Abstract— **The FOAF ontology has become a core model for describing social data of people and their links on the Web, and the people ranking is growing in popularity since search engines are considering the author's reputation of a Web page when generating search results. FOAF alone is yet insufficient to model social networks for ranking people on the Web. We concentrate here in co-authorship networks as a special case of social networks, and propose a model which extends FOAF with PageRank and AuthorRank metrics, both implemented in Semantic Web Rule Language (SWRL), for gauging the reputation of authors. Preliminary results are demonstrated, showcasing also the huge potential of this ranking approach for adopting it on social networks in general, and by search engines in particular where our future work will focus.**

Keywords-AuthorRank, Ranking algorithms on the Web, FOAF, Co-authorship networks, Social networks, SWRL rules, Ontology modeling

I. INTRODUCTION

With the "invasion" of social computing applications on the Web, Jim Handler and Tim Berners-Lee envision that we are just at the beginning of this growing and evolving age of "social machines" [1]. Social networks are mainly about linking people [2][3], and the FOAF ontology has become a core model on the Web for describing the profile data of people and their social links [4]. The focus on the Web has now shifted from the hypertext links (aka `href`) to people links (aka `knows`). Search engines are also shifting from ranking only pages, to ranking pages improved by considering the ranking of people authoring those pages. As Google and Bing announced, the reputation of authors of Web pages will influence the tops displayed in search results.

FOAF alone is yet insufficient to model social networks for ranking people on the Web. It has neither been designed to support ranking of people. Co-authorship networks make up an important subset of the large amount of data found in Web-based social networks.

Therefore, we concentrate here in co-authorship networks, and propose a model which extends FOAF with PageRank [5] and AuthorRank [6] metrics, both implemented in Semantic Web Rule Language (SWRL) [7][8], for gauging the reputation of authors. Actually, this extension relies on our earlier work [9]which already extended FOAF into CO-AUTHORONTO, but aimed to support social network analyses (SNA) metrics for co-authorship networks. According to [6], AuthorRank is an advanced version of PageRank which takes into account the *weight* of co-author links when ranking, i.e., it does not assume that all links have the same weight.

Referring further to [1], the growing Semantic Web provides necessary support for social computing technologies. Our framework exactly utilizes the Semantic Web technologies to support ranking in social networks, like the co-authorship networks.

The rest of the paper is organized as follows: We first introduce related work, followed by the background knowledge required to understand the rest of the paper, where CO-AUTHORONTO, our earlier extension of FOAF to capture the semantics of weighted co-authorship networks is also covered. Further we detail our contribution: the extension of FOAF, namely of CO-AUTHORONTO, and rule definitions in SWRL, to rank authors by PageRank and AuthorRank. Then an evaluation of our ranking framework is presented. Finally we conclude by highlighting the main contribution of this work, and the future work to follow.

II. RELATED WORK

In [10], a study to rank people based on FOAF data (referred to as AuthorRank there), and combining people ranking with co-citation analysis is presented. Ranking has been carried out by storing and querying in Yars [11], a Semantic Web platform. Another framework [12] deploys Semantic Web for the representation and people ranking by centrality measures of SNA expressed in form of SPARQL queries. Applying SNA metrics to the co-authorship networks on the Web has been investigated in [13][14] in finding interesting facts about authors of scientific publications and relationships among them. [6] introduces ranking co-authorship networks, including their ranking by PageRank and AuthorRank.

To the best of our knowledge, there is no approach to date which is based solely on the Semantic Web to rank people in social networks or co-authorship networks based on PageRank and AuthorRank. Nor is there an approach of considering (SWRL) rules of Semantic Web to express and reason over PageRank and AuthorRank for ranking people (authors) in co-authorship networks.

III. BACKGROUND

Next, a set of preliminary knowledge required for the rest of the work is briefly introduced:

- Definition of ranking measures like PageRank and AuthorRank for authors in co-authorship networks.
- Modeling co-authorship networks on the Web using CO-AUTHORONTO, an extended FOAF ontology.
- SNA metrics like exclusivity, frequency, or weight, and their implementation for co-authorship networks.

A. *Author Ranking via PageRank and AuthorRank*

Let us first recall the definitions according to [6] for PageRank and its weighted version, AuthorRank, of the directed weighted co-authorship graph model we adopt to rank authors in co-authorship networks on the Web. Let $A = \{a_1, .., a_k, .., a_n\}$ denote the set of n authors.

Definition 1 (PageRank). The PageRank of an author a_i is given as follows:

$$PR(a_i) = (1 - d) + d \sum_{j=1}^{n} \left(PR(a_j) \cdot \frac{1}{C(j)} \right)$$

where $PR(a_j)$ corresponds to the PageRank of the backlinking co-author node, and $C(j)$ is defined as the number of links going out of co-author j. In the first initial iteration of the PageRank algorithm, it is assumed that $PR(a_j) = 1$ for $j = 1..n$.

Definition 2 (AuthorRank). The AuthorRank of an author a_i is given as follows:

$$AR(a_i) = (1 - d) + d \sum_{j=1}^{n} (AR(a_j) \cdot w_{j,i})$$

where $AR(a_j)$ corresponds to AuthorRank of the backlinking co-author node, and $w_{j,i}$ to the edge weight between co-authors a_j and a_i. Analogous to the initial PageRank, the first initial iteration of the AuthorRank algorithm assumes $AR(a_j) = 1$ for $j = 1..n$.

One may think of AuthorRank as a generalization of PageRank by substituting $1/C(j)$ with $w_{j,i}$. According to [6], AuthorRank better reveals status of actors than centrality SNA measures and PageRank.

Example 1. The running example we will use throughout this paper consists of two publications and their respective authors as follows:

Publication	Authors		
First Course in Database Systems	Ullman	Widom	
Active Database Systems		Widom	Ceri

Fig. 1 provides the metrics calculated following the definitions 1 and 2 for the example above.

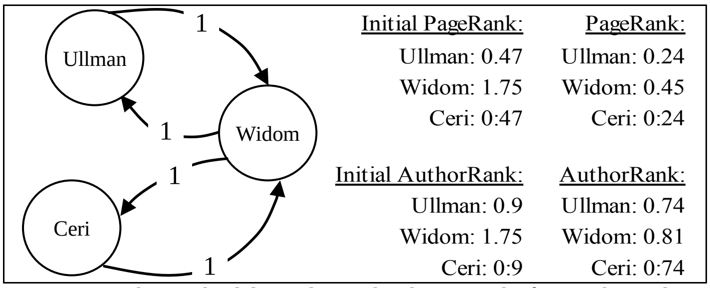

Figure 1. The weighted directed co-authorship network of *Example 1* and ranking values

B. *CO-AUTHORONTO: a FOAF Extension to Model Co-authorship Networks on the Web*

In our earlier work [9], we extended FOAF into CO-AUTHORONTO (Fig. 2) to model rich co-author relations with attributes such as *weight*, as well as other knowledge relevant to collaboration schemes in the scientific community, as suggested in [6]. In addition to FOAF, the CO-AUTHORONTO integrates the AKT Reference Ontology [15], its Portal and Support sub-ontologies describing people, publications, etc.

Two main classes in the CO-AUTHORONTO ontology are `Person` and `Publication`, with their respective subclasses, such as `Researcher`, or `Student` for the former, or, `Serial-Publication`, or `Book` for the later, all inherited either from the AKT ontologies, or FOAF.

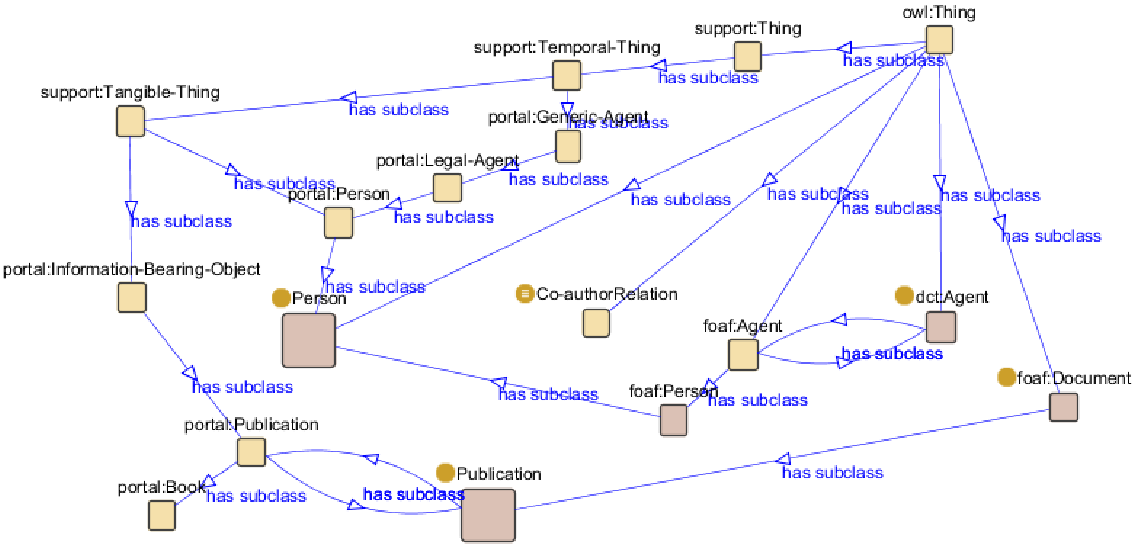

Figure 2. An excerpt of the CO-AUTHORONTO ontology – how it relates to the `foaf`, `portal` and `support` (class hierarchy) ontologies.

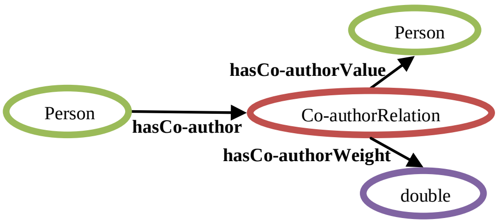

Figure 3. N-ary `hasCo-author` property in CO-AUTHORONTO (class view).

Each `Publication` instance may have (multiple) values of type `Person` on the `hasAuthor` property. The `hasExclusivity` property stores the exclusivity degree on a given publication, as explained in Tab.1. All co-author pairs of the same publication should share a common value for the exclusivity. The property `nrAuthors`, as its naming reveals, is designed to record data about the number of authors of a given publication. For `hasExclusivity` and `nrAuthors`, their values are inferred once their corresponding rule implementations in SWRL (Tab.1) are evaluated.

Fig. 3 illustrates the use of OWL n-ary relations [16] for representing the `hasCo_author` property of the `Person` class. It links an individual `Person` to another individual `Person`, as well as to another value, *weight*, which characterizes that `hasCo-author` relation.

C. SNA Metrics on Co-authorship Networks on the Web

Let again $A = \{a_1, .., a_k, .., a_n\}$ denote the set of n authors, whereas $P = \{p, .., p_k, .., p_m\}$ the set of m publications. Let $f(p_k)$ define the number of authors of publication p_k. Then in Tab. 1, definitions of SNA metrics, i.e., the exclusivity, co-authorship frequency, total frequency, and weight [6], as well as their respective rule definitions in SWRL according to our earlier work [9] are summarized. Once these metrics' rules are evaluated by an inference engine, their inferred values are assigned to the CO-AUTHORONTO ontology.

IV. RANK AUTHORS IN WEB CO-AUTHORSHIP NETWORKS

Now we describe in detail our proposal to further extend FOAF, namely CO-AUTHORONTO [9] which is originally built on top of FOAF and AKT ontologies and supports SNA metrics [6] just introduced in the previous section, to support PageRank and AuthorRank for ranking authors,. The main novelty of our proposal, i.e. the Semantic Web rule expressions for computing PageRank and AuthorRank are also introduced.

A. FOAF Extended with PageRank and AuthorRank

Let us next introduce our proposal to extend the CO-AUTHORONTO (Fig. 1) ontology [9] with new ontological constructs which enable ranking by PageRank and AuthorRank of authors in directed weighted co-authorship networks.

TABLE 1. SNA METRICS AND THEIR RULE DEFINITIONS IN CO-AUTHORONTO.

SNA metric		***SWRL rule expression***
Co-author relation	Not applicable	`Publication(?p) ∧ hasAuthor(?p,?a1) ∧ abox:hasURI(?a1,?a1URI) ∧ swrlb:substringAfter(?a1name,?a1URI,"#") ∧ hasAuthor(?p,?a2) ∧ abox:hasURI(?a2,?a2URI) ∧ swrlb:substringAfter(?a2name,?a2URI,"#") ∧ swrlb:notEqual(?a1name,?a2name) ∧ swrlx:makeOWLThing(?rel,?a1,?a2) → Co-authorRelation(?rel) ∧ hasCo-author(?a1,?rel) ∧ hasCo-authorValue(?rel,?a2)`
Number of authors of a given publication	$f(p_k)$	`Publication(?p) ∧ hasAuthor(?p,?a) ˚ sqwrl:makeSet(?s,?a) ∧ sqwrl:groupBy(?s,?p) ˚ sqwrl:size(?size,?s) → nrAuthors(?p,?size)`
Exclusivity per publication: Degree to which authors a_i, a_j have an exclusive co-authorship relation for a particular p_k publication. This definition gives more weight to co-author relationships in publications with fewer total co-authors than publications with large numbers of co-authors, i.e., it weighs the co-author relation in terms of how exclusive it is.	$g_{i,j,k} = \frac{1}{(f(p_k) - 1)}$	`Publication(?p) ∧ nrAuthors(?p,?nr) ∧ swrlb:greaterThanOrEqual(?nr,2) ∧ swrlm:eval(?e,"1/(nr-1)",?nr) → hasExclusivity(?p,?e)`
Co-authorship frequency: Sums up the exclusivity values $g_{i,j,k}$ for that same pair i,j of authors across all publications k (k=1..m) where they appear as co-authors a_i,a_j. This gives more weight to authors who co-publish more publications together, and do so exclusively.	$c_{ij} = \sum_{k=1}^{m} g_{i,j,k}$	`Publication(?p) ∧ nrAuthors(?p,?nr) ∧ swrlb:greaterThanOrEqual(?nr, 2) ∧ hasAuthor(?p,?a1) ∧ hasAuthor(?p,?a2) ∧ hasCo-author(?a1,?rel) ∧ hasCo-authorValue(?rel,?a2) ∧ abox:hasURI(?a1,?a1URI) ∧ abox:hasURI(?a2,?a2URI) ∧ swrlb:notEqual(?a1URI,?a2URI) ∧ sameAs(?a1,?a1) ∧ sameAs(?a2,?a2) ∧ hasExclusivity(?p,?e) ˚ sqwrl:makeSet(?s,?e) ∧ sqwrl:groupBy(?s,?a1,?a2) ˚ sqwrl:sum(?f,?s) → hasCo-authorFrequency(?rel,?f)`
Total co-authorship frequency: The sum of all co-authorship frequency values c_{ik} over a particular author ai and all of its co-authors a_k (k=1..n) in whatever publications where a_i appears as author	$c_i = \sum_{k=1}^{n} c_{ik}$	`Person(?a1) ∧ hasCo-author(?a1,?rel) ∧ hasCo-authorValue(?rel,?a2) ∧ hasCo-authorFrequency(?rel,?f) ˚ sqwrl:makeBag(?s,?f) ∧ sqwrl:groupBy(?s,?a1) ˚ sqwrl:sum(?totalFa1,?s) → hasTotalFrequency(?a1,?totalFa1)`
Co-author weight: Normalization by taking into account the total co-authorship frequency c_i of a given author a_i when calculating the co-authorship frequency c_{ij} between that author a_i and any other co-author a_j of him/her.	$w_{ij} = \frac{c_{ij}}{c_i}$	`hasCo-author(?a1,?rel) ∧ hasCo-authorValue(?rel,?a2) ∧ hasTotalFrequency(?a1,?totalFa1) ∧ hasCo-authorFrequency(?rel,?a12f) ∧ swrlm:eval(?w,"a12f/totalFa1",?a12f,?totalFa1) → hasCo-authorWeight(?rel,?w)`

The `Person` class is among extended concepts to support ranking, and its extension is defined as following:

```
Person   a  owl:Class;
    rdfs:subClassOf
    [a   owl:Restriction;
         owl:allValuesFrom: float;
         owl:onProperty: hasPR].
    rdfs:subClassOf
     [a   owl:Restriction;
         owl:allValuesFrom: float;
         owl:onProperty: hasAR];
    rdfs:subClassOf
    [a   owl:Restriction;
         owl:allValuesFrom: int;
         owl:onProperty:
          hasTotalNrCo-Authors].
```

- Added new datatype properties, `hasPR` and `hasAR`, both of domain `Person`, and their range set to `float`.
- Added a new datatype property `hasTotalNrCo-Authors`, of domain `Person` and the `int` range.

Further, a new class `Constants` is introduced:

```
:Constants   a  owl:Class;
      rdfs:subClassOf
      [a   owl:Restriction;
           owl:allValuesFrom: int;
           owl:onProperty:
                totalNrAuthors].
```

- Defined the datatype property `totalNrAuthors`, of domain `Constants`, and its range set to `int`.
- Instantiated a new single `Constants` individual, the `Constant1` individual, to hold the sole expected value for the `totalNrAuthors` property in our ontology (aka assigning a value to a global variable in programming languages).

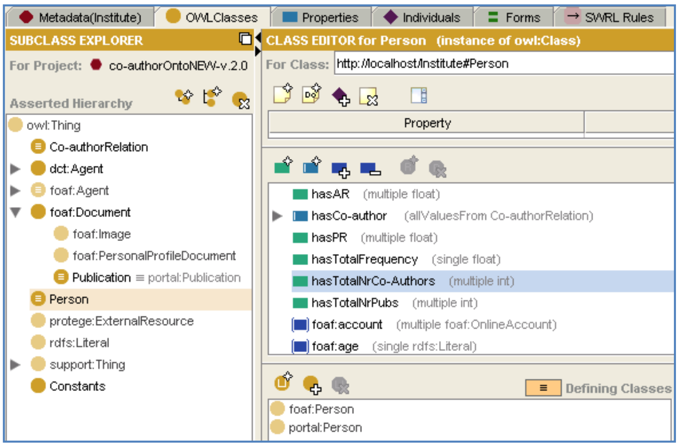

Figure 4. Some properties of the `Person` class introduced in our ontology.

Notice in Fig. 4 (a snapshot of our ontology in Protégé [17]) the class `Person` in the class hierarchy on the left window, its definition as equivalent class to classes `portal:Person` of AKT ontology, and `foaf:Person` of FOAF ontology shown in the right bottom window, and a list of its properties and restrictions on them shown in the right top window.

The `hasPR` and `hasAR` properties store the PageRank and AuthorRank values, respectively, for a given author of type `Person`. The `hasTotalNrCo-Authors` holds the total number of authors acting as co-authors for a given author in whatever publication in the ontology. The property `totalNrAuthors`, as its naming reveals, is designed to record data about the number of (distinct) authors in total for the whole ontology. For all these four properties, `hasPR`, `hasAR`, `hasTotalNrCo-Authors`, and `totalNrAuthors`, their values are inferred by reasoning through SWRL rules over the ontology, as will be introduced in the next section.

Fig. 5 illustrates an author, Ullman, and its values for the ranking-related properties. The `SWRLInjected_1` acts somewhat as an abstract relation instance: It serves merely to link from the source instance (Ullman) to this relation instance (`SWRLInjected_1`), and from this relation instance (`SWRLInjected_1`), to the target instance (Widom).

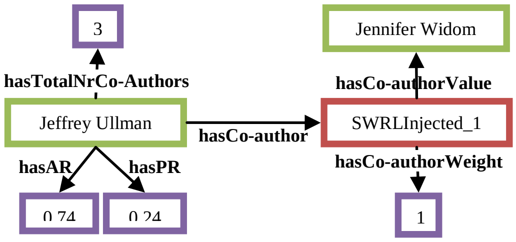

Figure 5. An instance view of the Person class and its ranking-related properties in our ontology: *Example 1* excerpt: Ullman and Widom are co-authors with weight 1; Ullman has PageRank value 0.24 (`hasPR`), AuthorRank value 0.74 (`hasAR`), and in total shares publications with 3 other authors in the ontology (`hasTotalNrCo-Authors`).

B. SWRL Rules to Calculate PageRank and AuthorRank

Moving from ontologies upwards the Semantic Web layer architecture, we will encounter the logic layer with rules which may capture semantics that cannot be directly captured using description logic of OWL [18].

We have defined rules in SWRL to implement PageRank and AuthorRank metrics [6] in conformance with the related ontological constructs we defined in the extended CO-AUTHORONTO ontology. Next we detail each of these rules and the metrics of the weighted co-authorship graph model they implement.

The values of these metrics, once pre-calculated through an inference engine, are assigned to the existing ontology annotations.

Rule 1 (Total number of authors). This rule will find the overall number of `Person` individuals in the ontology.

```
portal:Person(?a) ˚ sqwrl:makeSet(?s,?a) ˚
sqwrl:size(?n,?s) →
totalNrPersons(Constant1,?n)
```

It will collect all persons ?a, i.e., authors of all existing publications, into a common set ?s (the makeSet built-in predicate), and then extract the total number of elements, ?n, in the set (the size built-in predicate). The number obtained will then be asserted to the single existing totalNrPersons property of the Constant1 individual in the ontology.

Rule 2 (Total number of co-authors of a given author). Next, the rule in SWRL to calculate the number of co-authors for each given person is presented.

```
portal:Person(?a2) ∧ hasCo-author(?a2,?rel)
∧ hasCo-authorValue(?rel,?a1) ∧
abox:hasURI(?a1,?a1URI) ∧
abox:hasURI(?a2,?a2URI) ∧
swrlb:notEqual(?a1URI,?a2URI) ∧
sameAs(?a1,?a1) ∧ sameAs(?a2,?a2) ˚
sqwrl:makeBag(?s,?a1) ∧ qwrl:groupBy(?s,?a2)
˚ sqwrl:size(?allca,?s) →
hasTotalNrCo-Authors(?a2,?allca)
```

Analogous to Rule 1, the collection built-ins makeBag (bags allow duplicate elements, sets do not) and groupBy are applied, but this time targeted to collect co-author pairs, group them per author (person), and finally calculate the number of elements (the size built-in predicate), i.e., of co-authors' that belong to that given group. The result will assert a value on the hasTotalNrCo-Authors property of the given author individual ?a1, once per each such author (i.e., per group created). The notEqual built-in predicate does exclude a combination of an author with himself (a1, a1) as a possible co-author pair.

Rule 3 (Initial PageRank). Following Definition 1, a rule to calculate the first iteration of the PageRank over authors, i.e., over a given author ?ai is shown below, and proceeds as follows:

1. It considers all backlinking co-author relations to ?a1, e.g., that ?rel connecting author ?aj to author ?ai
2. Applies the sum built-in predicate to sum up into ?ally the contributions of all backlinking co-authors to ?ai, e.g., the contribution ?yj=1/?cj of co-author ?aj, where ?cj is the nr. of co-author relations going out of co-author ?aj;
3. Evaluates the PageRank formulae according to Definition 1 taking into account also the dumping factor which is set to 0.85, as well as the total number of authors in the ontology as provided by the totalNrPersons predicate (cf. Rule 2).

```
hasCo-author(?aj,?rel) ∧
hasCo-authorValue(?rel,?ai) ∧
abox:hasURI(?aj,?pjURI) ∧
abox:hasURI(?ai,?piURI) ∧
swrlb:notEqual(?ajURI,?aiURI) ∧
sameAs(?aj,?aj) ∧ sameAs(?ai,?ai) ∧
hasTotalNrCo-Authors(?aj,?cj) ∧
totalNrPersons(Constant1,?n) ∧
swrlm:eval(?yj,"1/cj",?cj) ˚
sqwrl:makeBag(?s,?yj) ∧ qwrl:groupBy(?s,?ai)
˚ sqwrl:sum(?ally,?s) ∧
swrlm:eval(?pri,"(0.15 / n) + 0.85 *
ally",?n,?ally) → hasPR(?ai,?pri)
```

The eval is just another built-in of the swrlm library of SWRL to evaluate the given mathematical formula.

Rule 4 (PageRank). Once the initial PageRank SWRL rule as defined above is evaluated, the system further evaluates the actual PageRank rule for each (other) author recursively. The only difference of this rule to the initial PageRank rule (Rule 3) is in step 2:

- The contributions of all backlinking co-authors to the author under consideration, e.g., to author ?ai, involve also the actual PageRank values of backlinking co-authors, e.g., the PageRank of co-author ?aj. In other words, the contribution ?yj=prj/?cj, where prj is the actual PageRank value of that co-author ?aj, and ?cj is the nr. of co-author relations going out of co-author ?aj. At the first iteration of this rule, the PageRank values at the body of the rule are those inferred by the initial PageRank rule.

```
hasPR(?aj,?prj) ∧ hasCo-author(?aj,?rel) ∧
hasCo-authorValue(?rel,?ai) ∧
abox:hasURI(?aj,?ajURI) ∧
abox:hasURI(?ai,?aiURI) ∧
swrlb:notEqual(?ajURI,?aiURI) ∧
sameAs(?aj,?aj) ∧ sameAs(?ai,?ai) ∧
hasTotalNrCo-Authors(?aj,?cj) ∧
totalNrPersons(Constant1,?n) ∧
swrlm:eval(?yj,"prj/cj",?prj,?cj) ˚
sqwrl:makeBag(?s,?yj) ∧ qwrl:groupBy(?s,?ai)
˚ sqwrl:sum(?ally, ?s) ∧
swrlm:eval(?pri,"0.15 / n + 0.85 *
ally",?n,?ally) → hasPR(?ai,?pri)
```

This rule will iterate once over all authors and their co-author pairs found in the ontology (the ?rel individuals of the Co-authorRelation class), compute their initial PageRank values, and accordingly augment them using the hasPR properties which represent PageRank.

Rule 5 (Initial AuthorRank). The SWRL rule which implements the initial AuthorRank is distinct from the SWRL rule which implements the initial PageRank in step 2 (cf. Rule 3), namely in:

- The contributions which it sums up into ?ally of all backlinking co-authors to ?ai. For instance, the contribution of co-author ?aj is ?yj=?wji, where ?wji is the weight of the co-author relation between the author ?ai under consideration and its current co-author ?aj.

```
hasCo-author(?aj,?rel) ∧
hasCo-authorValue(?rel,?ai) ∧
hasCo-authorWeight(?rel,?wji) ∧
abox:hasURI(?aj,?pjURI) ∧
abox:hasURI(?ai,?piURI) ∧
swrlb:notEqual(?ajURI,?aiURI) ∧
sameAs(?aj,?aj) ∧ sameAs(?ai,?ai) ∧
sqwrl:makeBag(?s,?wji) ∧
sqwrl:groupBy(?s,?ai) ˚ sqwrl:sum(?sumj,?s)
∧ totalNrPersons(Constant1,?n) ∧
swrlm:eval(?ari,"(0.15 / n) + 0.85 *
sumj",?n,?sumj) → hasAR(?ai,?ari)
```

This corresponds to exactly substituting what was `1/C(j)` in intitial PageRank with $w_{j,i}$ in initial AuthorRank.

Rule 6 (AuthorRank). Finally, the following rule calculates the AuthorRank (Definition 2) of every author in the ontology given: (1) the co-authorship network, and (2) the initial AuthorRank values of its co-authors calculated by Rule 5. It reasons over values assigned to the `hasCo-authorWeight` property of a particular `?rel` co-author relationship between `?aj` and `?ai`, uses the `makeBag` built-in predicate to collect them into a bag rather than a set (bags allow duplicate values, whereas sets do not), groups the bag values by author `?ai`, and calculates the sum of values on each group separately, yielding thus the value for AuthorRank of the respective author (bound to that respective author on his `hasAR` property in our ontology).

```
hasCo-author(?aj,?rel) ∧ hasCo-
authorValue(?rel,?ai) ∧ hasCo-
authorWeight(?rel,?wji) ∧
abox:hasURI(?aj,?pjURI) ∧
abox:hasURI(?ai,?piURI) ∧
swrlb:notEqual(?ajURI,?aiURI) ∧
sameAs(?aj,?aj) ∧ sameAs(?ai,?ai) ∧
hasPR(?aj,?arj) ∧ swrlm:eval(?yj, "arj*wji",
?arj,?wji) ˚ sqwrl:makeBag(?s,?yj) ∧
sqwrl:groupBy(?s,?ai) ˚ sqwrl:sum(?sumj,?s)
∧ totalNrPersons(Constant1, ?n) ∧
swrlm:eval(?ari,"(0.15)/n + 0.85 *
sumj",?n,?sumj) → hasAR(?ai,?ari)
```

As one may notice, the only difference of this rule to the initial AuthorRank rule (cf. Rule 5) is following:

- The contributions of all backlinking co-authors to the author under consideration, e.g., to author `?ai`, involve also the actual AuthorRank values of backlinking co-authors, e.g., the AuthorRank of co-author `?aj`. In other words, the contribution `?yj=arj*wji`, where `arj` is the actual AuthorRank value of that co-author `?aj`, and `?wji` is the weight of the co-authorship relation between co-authors `?aj` and `?ai`. At the first iteration of this rule, the AuthorRank values at the body of the rule are those inferred by the initial AuthorRank rule.

V. EVALUATION RESULTS

The analyses run by ORA [19], a social network analysis tool, demonstrates there is a dense 3D cloud of 12482 nodes and 10984 co-author edges in our test-bed of data (Fig. 6).

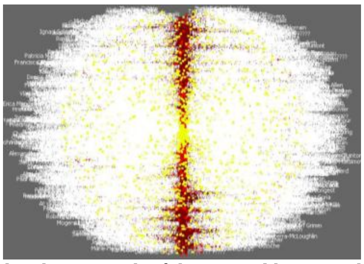

Figure. 6. Co-authorship network of the SEEU library: authors (in red), co-authors (in yellow), and their links (in white).

Our approach has first been tested on the small sample example, referred to as the running example (Example 1) in this work. Next we show the evaluation of our ranking SWRL rules over a real-world co-authorship network. Experiments have been conducted against the co-authorship data we generated from the library database of the South East European University (SEEU) which stores the bibliography of books and other publications available for loan from the university. It contains 12774 `Publication` individuals, and 11607 `Person` individuals acting as authors of publications. The experiment was run on a 2 x Quad Core 2.5 GHz machine with 16 GB RAM and Windows Server 2003 2R installed. Centrality measures generated with ORA of the same co-authorship network are given in Tab. 2.

TABLE 2. TOP 10 RANKED AUTHORS BY THEIR CENTRALITY MEASURES.

Nr	betweenness centrality	closeness centrality	degree centrality
1	Baki Ymeri	William Shakespeare	Ilia Kristo
2	Ralph Mowat	Bardhyl Ceku	William Shakespeare
3	William Shakespeare	Arjan Abazi	Theo Scherling
4	David Cotton	Artan Duka	Miroslav Hadzic
5	Kenneth E Clow	Mustafa Ibrahimi	Clare West
6	Ali Aliu	Clare West	Rebecca Hughes
7	Anne Frank	Miso Nikolov	Ismail Kadare
8	Richard M Hodgetts	Gadaf Rexhepi	Emily Dickinson
9	Gabriel Almond	Ralph Mowat	Erion Kristo
10	Penelope Lively	Theo Scherling	Diane Mowat

The SWRL rules in our framework did infer that there are 12732 co-author relationships (the total number of `hasCo-authorValue` properties in our ontology) as a special type of the FOAF `knows` relationship. Next we present results of our benchmarks on ranking metrics, PageRank and AuthorRank, calculated in our framework of SWRL rules over the extended CO-AUTHORONTO representation of the SEEU library co-authorship network.

Query 1 (PageRank). The SQWRL query expression below is used to find the PageRank values of authors in the SEEU library as defined by Rule 4 earlier in our system.

```
hasTotalFrequency(?a1,?tFa1) ∧ hasTotalNrCo-
Authors(?a1,?tNrCa1) ∧ hasPR(?pra1) ∧
portal:full-name(?a1,?n1) →
sqwrl:columnNames("author","name","nr co-
authors","total co-authorship frequency",
"PageRank") ∧ sqwrl:select(?a1,?n1,?tNrCa1,
```

```
?tFa1,?pra1) ∧ qwrl:orderByDescending(?pra1)
```

The result of this query is a table of cardinality 7006 with top 10 authors on their PageRank values as shown in Tab. 3.

As one may also reason from the list in Tab. 3, there are cases when an author, e.g. Ismail Kadare, has more co-authors (might share diverse roles, like writer, editor, etc.), as well as more exclusive co-author relations (the higher total co-authorship frequency), but has lower PageRank and is hence ranked lower in our table (position 4) than is, e.g. Diane Mowat (position 3) with a higher score for PageRank. Thus, a sum of less but more exclusive co-author relations may outperform a higher in number but weaker in exclusivity co-author relations.

TABLE 3. TOP 10 RANKED AUTHORS BY PAGERANK.

nr	author	name	nr co-authors	total co-authorship frequency	PageRank
1	#P21542	William Shakespeare	24	18.5	14.102285
2	#P22253	Clare West	19	18	9.3027906
3	#P22263	Diane Mowat	14	12	8.5708466
4	#P21104	Ismail Kadare	17	14.5	8.2166796
5	#P22997	Honoré de Balzac	12	10	7.5083461
6	#P22463	Ali Aliu	12	12	7.4658461
7	#P31020	Erion Kristo	12	9	7.3666797
8	#P23001	Rex Gibson	11	9.5	6.8354297
9	#P20527	Noam Chomsky	13	8	6.8000131
10	#P22235	John Escott	12	11	6.7433462

Query 2 (AuthorRank). A similar SQWRL query expression is used to check the AuthorRank values of authors in the SEEU library defined by Rule 2 in the previous section.

```
hasTotalFrequency(?a1,?tFa1) ∧ hasTotalNrCo-
Authors(?a1,?tNrCa1) ∧ hasAR(?ara1) ∧
portal:full-name(?a1,?n1) →
sqwrl:columnNames("author","name","nr co-
authors","total co-authorship frequency",
"AuthorRank") ∧
sqwrl:select(?a1,?n1,?tNrCa1,?tFa1,?ara1) ∧
sqwrl:orderByDescending(?ara1)
```

This query's result proved that, different from PageRank, ranking by AuthorRank is sensitive to the co-author weights.

VI. CONCLUSION AND FUTURE WORK

FOAF has not been designed to support ranking of people. We extended the FOAF ontology with PageRank and AuthorRank metrics: the `hasPR` and `hasAR` properties for each Person (author) individual in the co-authorship network. What is an extreme contribution, we consider, is our innovative approach of designing and implementing the PageRank and AuthorRank algorithms using rules in Semantic Web. Although still work remains to advance this approach to further complex ranking scenarios, our preliminary evaluation results show that there is a promising future to apply that same approach over a wealth of social data on the WWW and their growing in number "social machines".

Our framework may easily be adopted to support ranking social networks in general, given the SWRL rules are then applied to rank people using AuthorRank based on their weighted co-authorship or any other weighted relation, or the reputation gained following, e.g., their `knows` relationships, and the PageRank algorithm instead. The people ranking is growing in popularity since search engines are considering the author's reputation of a Web page when generating search results. Think of Google+ and the FOAF mechanism "behind the scenes" to support author ranking. This is where our future work will concentrate: rank search result of Web pages by considering the reputation of authors of those pages, all supported solely by Semantic Web technologies.

REFERENCES

[1] Hendler, J., & Berners-Lee, T. (2010). From the Semantic Web to social machines: A research challenge for AI. *Artif. Intell. , 174* (2).

[2] Goldbeck, J. (2007). The dynamics of Web-based social networks: Membership, relationships, and change. *First Monday , 12* (11).

[3] Goldbeck, J. (2005). *Web-based Social Networks: A Survey and Future Directions.* Technical Report, University of Maryland.

[4] Brickley, D., & Miller, L. (2010, January 1). *FOAF Vocabulary Specification 0.97.* Retrieved June 2010, 28, from http://xmlns.com/foaf/spec/20100101.html

[5] Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The PageRank Citation Ranking: Bringing Order to the Web.* Stanford InfoLab.

[6] Liu, X., Bollen, J., Nelson, M. L., & Van de Sompel, H. (2005). Co-authorship networks in the digital library research community. *Inf. Process. Manage. , 41* (6), 1462--1480.

[7] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004, May 21). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML.* Retrieved June 27, 2010, from http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/

[8] O'Connor, M. (2010, February 12). *SWRLLanguageFAQ.* Retrieved from http://protege.cim3.net/cgi-bin/wiki.pl?SWRLLanguageFAQ

[9] Ahmedi, L., Abazi-Bexheti, L., & Kadriu, A. (2011). A Uniform Semantic Web Framework for Co-authorship Networks. *IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing* (pp. 958-965). Sydney, Australia: IEEE.

[10] Ding, Y., Scharffe, F., Harth, A., & Hogan, A. (2006). AuthorRank: Ranking Improvement for the Web. *Proc. of the Int. Semantic Web and Web Services Conference (SWWS'06).*

[11] Harth, A., & Decker, S. (2005). Optimized Index Structures for Querying RDF from the Web. *Proceedings of the Third Latin American Web Congress* (pp. 71-). IEEE Computer Society.

[12] Erétéo, G., Buffa, M., Gandon, F., & Corby, O. (2009). Analysis of a Real Online Social Network using Semantic Web Frameworks. *8th International Semantic Web Conference, ISWC'09.* Wiley Publishing, Inc.

[13] De Castro, R., & Grossman, J. W. (1999). Famous trails to Paul Erdös. *MATHINT: The Mathematical Intelligencer , 21*, 51--63.

[14] Nascimento, M. A., Sander, J., & Pound, J. (2003). Analysis of SIGMOD's co-authorship graph. *SIGMOD Rec. , 32* (3), 8--10.

[15] *The AKT Reference Ontology*. (2003). Retrieved June 23, 2011, from (AKT), Advanced Knowledge Technologies: http://www.aktors.org/publications/ontology/

[16] Noy, N., & Rector, A. (2006, April 12). *Defining N-ary Relations on the Semantic Web.* Retrieved June 27, 2010, from http://www.w3.org/TR/2006/NOTE-swbp-n-aryRelations-20060412/

[17] Knublauch, H., Fergerson, R. W., Noy, N. F., & Musen, M. A. (2004). The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. *3rd International Semantic Web Conference, ISWC'04. 3298*, pp. 229-243. Springer LNCS.

[18] McGuinness, D. L., & van Harmelen, F. (2004, February 10). *OWL Web Ontology Language Overview.* Retrieved June 27, 2010, from http://www.w3.org/TR/2004/REC-owl-features-20040210/

[19] Carley, K. M. (2001-201). ORA Project. Pittsburg, USA.