ISWC2005

International Semantic Web Conference
November 6-10 2005
Galway, Ireland

http://iswc2005.semanticweb.org

W8: Semantic Network Analysis

Organisers: Gerd Stumme, Bettina Hoser, Christoph Schmitz, Harith Alani

Monday, 7 November 2005

ISWC 2005 could not take place without the generous support of the following sponsors

Super Emerald Sponsors

Gold Sponsors

Silver Sponsors

ISWC 2005 Organising Committee

General Chair	Mark Musen, Stanford University
Research Track Co-Chair	Yolanda Gil, Information Sciences Institute
Research Track Co-Chair	Enrico Motta, The Open University
Industrial Track Chair	V Richard Benjamins, iSOCO, S.A.
Workshop Chair	Natasha F Noy, Stanford University
Tutorial Chair	R.V. Guha, Google
Poster & Demo Chair	Riichiro Mizoguchi, Osaka University
Semantic Web Challenge	Michel Klein, Vrjie Universiteit Amerdam
Semantic Web Challenge	Ubbo Visser, Universitat Bremen
Doctoral Symposium Co-Chair	Edward Curry, National University of Ireland, Galway
Doctoral Symposium Co-Chair	Enda Ridge, University of York
Meta-Data Chair	Eric Miller, W3C
Sponsorship Chair	Liam O'Móráin, DERI Galway
Local Organising Co-Chair	Christoph Bussler, DERI Galway
Local Organising Co-Chair	Stefan Decker, DERI Galway
Local Organiser	Brian Cummins, DERI Galway
Webmaster	Seaghan Moriarty, DERI Galway
Web Design	Johannes Breitfuss, DERI Innsbruck

ISWC 2005 Workshop on

Semantic Network Analysis Workshop (SNA'05)

Monday, November 7, 2005 at Galway, Ireland

Organising Committee

Gerd Stumme
Universität Kassel
Fachbereich Mathematik/Informatik
Fachgebiet Wissensverarbeitung
Wilhelmshöher Allee 73
34121 Kassel
Email: stumme@cs.uni-kassel.de

Bettina Hoser
Lehrstuhl für Informationsdienste und elektronische Märkte
Universität Karlsruhe (TH)
Gebäude 20.20 (Rechenzentrum), Raum 164
D-76128 Karlsruhe, Germany
Email: bettina.hoser@em.uni-karlsruhe.de

Christoph Schmitz
Universität Kassel
Fachbereich Mathematik/Informatik
Fachgebiet Wissensverarbeitung
Wilhelmshöher Allee 73
34121 Kassel
Email: schmitz@cs.uni-kassel.de

Harith Alani
Intelligent, Agents, Multimedia Group
Electronics and Computer Science Dept.
University of Southampton
Highfield, Southampton, UK
Email: ha@ecs.soton.ac.uk

Programme Committee

Lada Adamic (HP Labs)

Vladimir Batagelj (University of Ljublijana)

Stefan Bornholdt (University of Bremen)

Ulrik Brandes (University of Konstanz)

John Davies (BT Exact)

Patrick Doreian (University of Pittsburg)

Tim Finin (University of Maryland)

Stéphane Laurière (Mandrake)

Nick Kings (BT Exact)

Sebastian Kruk (DERI Galway)

Kieron O'Hara (University of Southampton)

Nigel Shadbolt (University of Southampton)

Steffen Staab (University of Koblenz)

Rudi Studer (University of Karlsruhe)

Andrew Tomkins (IBM Almaden Research Center)

Andreas Hotho (University of Kassel)

Introduction

During the past years a shift in the fundamental understanding of the aims of Computer Science, especially in AI, could be observed. While early research in AI aimed at replacing the human being with better tools, the prevalent current vision is nowadays to support him in his tasks. This shows up in the rise of research areas like **communities of practice**, **knowledge management**, **web communities**, and **peer to peer**. In particular the notion of collaborative work - and thus the need of its systematic analysis - becomes more and more important.

On the other hand, techniques for analyzing such structures have a long tradition within sociology. While in the beginnings, researchers in that area had to spent huge efforts in collecting data, they nowadays often come for free in the WWW. Popular examples are citation and co-author graphs, friend of a friend etc.

Thus there exists an increasing interest of the social network analysis community in the web. The semantic web provides an additional aspect as it distinguishes between different kinds of relations, allowing for more complex analysis schemes.

Our aim is to bring the two communities together in order to learn from each other. We expect especially that the semantic web community can largely benefit from the long tradition present in social network analysis.

Besides analyzing social networks and cooperative structures within the (semantic) web, our second aim is to exploit the results for supporting and improving communities in their interaction. An important research topic is thus how to include network analysis tools in working environments such as knowledge management systems, peer to peer systems or knowledge portals.

Table of Contents

Full Papers

Position Paper

Invited Talk

A Case Study on Emergent Semantics in Communities*

Elke Michlmayr

Women's Postgraduate College for Internet Technologies (WIT),
Institute for Software Technology and Interactive Systems
Vienna University of Technology, Austria
michlmayr@wit.tuwien.ac.at

Abstract. This paper delivers a case study on the properties of meta-data provided by a folksonomy. We provide the background about folksonomies and discuss to which extend the process of creating meta-data in a folksonomy is related to the idea of emergent semantics as defined by the IFIP 2.6 Working Group on Data Semantics. We conduct experiments to analyse the meta-data provided by the *del.icio.us* folksonomy and to develop a method for selecting subsets of meta-data that adhere to the principle of interest-based locality, which was originally observed in peer-to-peer environments. In addition, we compare data provided by *del.icio.us* to data provided by the DMOZ taxonomy.

Keywords: Emergent Semantics in Communities, Folksonomies, Online harvesting of Semantic Network Information
Tags: folksonomies p2p computer science

1 Introduction

Recently, lots of discussions were raised by the advent of a new user-centric approach to categorization called folksonomies. Most of the debate is focused on the relationship between folksonomies and other approaches to categorization, such as taxonomies or ontologies. Folksonomies are comprised of a large amount of publicly available meta-data about lots of items, e.g., bookmarks or images, which can be retrieved from a central server. These meta-data are created by the users of the system without any restrictions posed by the system. Hence, the meta-data are inconsistent by nature, but the system tolerates these inconsistencies and exploits them for computing similarities between the keywords used for annotation. Our interest in folksonomies arises from being in need of simulation data for peer-to-peer applications. The contribution of this work is twofold: First, we deliver an in-depth study of the properties of meta-data produced by

* This research has partly been funded by the Austrian Federal Ministry for Education, Science, and Culture, and the European Social Fund (ESF) under grant 31.963/46-VII/9/2002.

folksonomies. Second, we investigate how meta-data produced by folksonomies can serve as simulation data for peer-to-peer environments.

The paper is organized as follows. In Section 2 we provide the necessary background about folksonomies and discuss their strengths and weaknesses. In Section 3 we compare the behaviour of the participants in a folksonomy to that of peers in a peer-to-peer network, and we examine the relationship between emergent semantics and folksonomies. In Section 4 we describe the experiments conducted on the provided meta-data in order to analyse its properties, and we report on the results of these experiments. Finally, in Section 5, we sum up our findings.

2 Folksonomies and their characteristics

The term *folksonomies* refers to a class of multi-user applications that provide a simple categorization system. This system is used to organize items, e.g., bookmarks or images. Instead of managing them within the browser application or on the local hard disk, the items are sent to a central server and stored there, together with meta-data authored by the user. These meta-data are comprised of one or more keywords — so-called *tags* — which describe the item. The keywords can be chosen freely by the user. Unlike in other categorization systems, there is no controlled vocabulary that defines which terms can be used as keywords in the categorization process. Another difference to existing categorization systems is that all keywords lie within the same namespace. There is no intention and hence no possibility to build hierarchical relationships between different keywords. The system uses a very simple data model which is depicted in Figure 1.

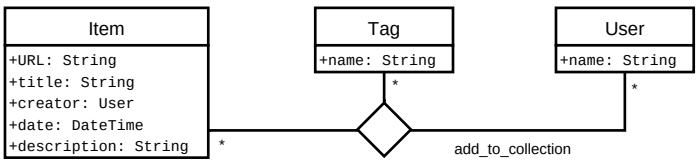

Fig. 1. Simple data model

The service provides each participant with his or her own Web page that shows the participant's item collection which contains all items together with the corresponding tags. The items are sorted in chronological order, showing the latest entry first. It is possible to filter the list of items by tag names to show only items that are annotated with a certain tag A. It is also possible to use another tag B to filter this list and retrieve all items that were annotated with both A and B.

These services are of value to the individual for managing items, but more important is that all participants of a folksonomy cooperate by allowing public

access to their item collection and the associated meta-data. Public availability has two advantages. First, all Web users have access to the annotated item collections. Second, since all meta-data are stored at one single server, it is possible to analyse and aggregate it without having additional communication costs in terms of bandwidth. Aggregations are performed for both items and tags. Information about items can be aggregated since the participants individually and independently create meta-data about the same items. For each item, there exists a Web page where all tags that an item was annotated with are shown, together with the total number of participants that used that tag to annotate the item (see Figure 2). In addition, the service lists all participants that included the item in their collection and shows the total number of these participants. The total number shows the popularity of a certain item. Aggregation for tags is possible because the participants use the same tags to annotate different items. For each tag, there exists a Web page with a list of all items at least once annotated with this tag. The result of these aggregations is a network of related concepts. At the data level, folksonomies are undirected weighted graphs that can be seen from different perspectives. When viewing items as the nodes of the network, two nodes are connected if they are annotated with the same tags. These kinds of connections are weighted. The more often the same tags were used, the higher the weight of the edge. When viewing the participants as the nodes of the network, the edges are built by those items that are shared between the participants. These graphs are exploited as input for an algorithm that computes the relatedness of tags. Each Web page containing aggregated information about a certain tag also shows the tag names of related tags as computed by the algorithm. For example, the algorithm used by the service *Flickr* [14] computes *tent*, *fire*, *hiking* as related tags for tag *camping*.

43 physics	15 mathematics	7 library	5 article	3 ai
41 science	10 articles	6 preprint	4 computer	3 study
27 research	10 journal	6 books	4 arxiv	
23 math	9 archive	6 programming	4 literature	
19 papers	8 biology	5 cs	4 toread	
18 reference	7 eprint	5 academic	4 computerscience	

Fig. 2. Tag distribution used to annotate a sample del.icio.us item

This approach to categorization is different to the top-down approach that is employed in traditional categorization systems, e.g. taxonomies or ontologies. Ontologies provide domain-specific vocabularies that describe the conceptual elements of a domain and the relationship between these elements, such as *is-a*-relationships or *part-of*-relationships. Creating an ontology requires careful analysis of what kind of objects and relations can exist in that domain [5]. This analysis is done by domain experts together with information architects who need to reach consensus about the exact meaning of objects and relations. After the ontology has been developed and put in place, the data items are categorized

according to the chosen categorization scheme. The same top-down approach is used in the database world. Before the actual data comes in, a schema is built that for that data. The schema defines which entities exists and how they are related. A third example is object-oriented modelling, where an instance can not exist without being a member of a class. The model defines the hierarchy of objects and their relationships. In a folksonomy, each participant uses a certain tag with his or her personal meaning in mind. There is no controlled vocabulary or ontology that defines the meaning of tags. Everybody has the possibility to express his or her opinions about the categorization of a certain object. This freedom of choice induces all problems controlled vocabularies try to avoid. The participants can use either the singular or the plural form of a term. Hence, they create two different tags with exactly the same meaning. There is no synonym control; hence different terms that refer to the same concept are in use. A special problem are keywords that consist of two terms: some participants create one tag by combining the terms with underscores or hyphens, or by creating a compound word with no separating character in between, while others create two tags, one for each term. The bottom-up approach to categorization avoids the necessity to reach consensus about the most appropriate categorization of a certain object. Semantic reconciliation is performed by the magnitude of participants that added meta-data to the folksonomy. The tags that are used most often to annotate a certain item express the opinion of the majority. Thus, the utility of a folksonomy is directly proportional to its number of participants and the amount of meta-data produced by them.

In addition to the already mentioned shortcomings of folksonomies that are caused by not using controlled vocabularies, a major weakness lies in the user interface. While browsing and filtering items by tag names is supported very well, searching for a certain item by name is impossible. Another shortcoming of folksonomies is that they can easily be spammed. Malicious participants can abuse the system by adding items of their interests and by assigning lots of tags for these items. There are a number of existing services like *del.icio.us* [8] for bookmarks, *Flickr* [14] for images, *Connotea* [9] for references to scientific literature, and others. A review of the existing services is provided in [7]. Services for bookmarks are also called *social bookmarking services*. Those services provide convenient interfaces to their participants, e.g. the possibility to add an item using a special link containing JavaScript code — a so-called bookmarklet — that transfers the URL to be added to the server. While typing in keywords, the participant is shown a list of the keywords he or she already used before. If the bookmark is already stored in the system, a list of popular keywords for that bookmark is shown as well. Those two lists are facilities that assist the participants in choosing appropriate keywords. After storing the bookmark, he or she is automatically redirected to the newly added Web page. More information about the general ideas behind folksonomies can be found in [3] and [7]. In [6], the major differences between folksonomies and taxonomies are discussed and some statistical information about the tag distribution of *del.icio.us* [8] is presented.

A case study on the service *Connotea* that provides a folksonomy for sharing meta-data about scientific literature can be found in [9].

3 Folksonomies and peer-to-peer environments

In Section 3.1, we explain why folksonomies can be used for retrieving simulation data for peer-to-peer networks. In Section 3.2, we briefly introduce the idea of emergent semantics and its relationship to folksonomies.

3.1 User behavior

Folksonomies are centralized services that heavily rely on the aggregation of meta-data. These aggregations are possible because the meta-data reside on a single server. For example, it is easy to determine all participants who share a certain information item. In a peer-to-peer environment, there is no central server and all peers store their information items at the local hard disk. Aggregating data consumes network resources. Knowing which peers share a certain information item is a non-trivial task for which it is necessary to track how the items are replicated within the network [4].

Although the architectures of folksonomies (centralized) and peer-to-peer networks (distributed) are completely different, the important point is that the behaviour of participants in a folksonomy is comparable to the behaviour of peers in an unstructured peer-to-peer network. All participants act autonomously and there is no central authority coordinating them. All participants provide information items to others that can be browsed and retrieved. Since the meta-data produced by folksonomies are publicly available and can be easily retrieved from one central server, folksonomies are suitable for retrieving test data for peer-to-peer applications. The available data can be used for modelling peers and their content distribution. Folksonomies do not provide any data about queries and query distribution. As a by-product of this paper, the data gathered during the experiments described in Section 4 will be used as a test suite for an algorithm for query routing in peer-to-peer networks described in [10].

3.2 Do folksonomies provide emergent semantics?

The term *emergent semantics* was defined by Aberer et al. in [2]. In this work, the authors discuss semantic interoperability for loosely coupled information sources and observe that a-priori agreements on concepts, e.g., the use of ontologies, are not appropriate in ad-hoc situations, because there is no possibility for the communicating peers to anticipate all interpretations. Instead, a semantic handshake protocol is suggested that allows negotiations between pairs of peers to reach an agreement over the meaning of models, e.g., by local schema mapping [1]. In order to save network resources, these negotiations are local interactions whenever possible. Global agreements are obtained by aggregating local agreements.

Semantic interoperability is constructed incrementally by lots of negotiations which are influenced by the context of existing global agreements.

Letting aside the major differences that stem from the fact that folksonomies operate in a centralized environment, there are some ideas from emergent semantics that can be found in folksonomies. As suggested in [2], folksonomies are self-organized systems. Both approaches (1) do not force their users to commit themselves to an existing ontology, (2) rely on lots of small interactions as well as on (3) aggregation of the results of these interactions, and both (4) construct their global properties incrementally. Another main distinction is that while the interactions in emergent semantics are initiated in order to reach consensus, in a folksonomy there are only information-exchanging acts that to not lead to a definite agreement. In summary, folksonomies employ an approach that is similar to emergent semantics, but address a simplified problem because a centralized architecture exists and because all participants rely on the same schema.

4 Experiments and Results

In this section we report on the experiments conducted on data retrieved from a social bookmarking service. In Section 4.1, the experimental setup is described. In Section 4.2, we show a method for data selection and present statistics about the retrieved test data sets. In Section 4.3 we evaluate if it is possible to join test data sets. In Section 4.4 we compare two different sets of categorization data for the same items. Finally, in Section 4.5 we analyse the impact of a bookmark's popularity.

4.1 Experimental setup and test data

The test data used in the following experiments was gathered by downloading[1] selected bookmarks from *del.icio.us* [8], which is one of the most successful social bookmarking services having more than 55.000 users. For the implementation of the downloading routines, Perl scripts were used. We kept a list of all already retrieved URLs to prevent multiple downloading of the same information. In addition, error handling facilities were necessary since internal server errors of the service occurred frequently. In order not to take up too many resources from the service, we used a delay of five seconds between each subsequent request. All downloaded data was saved to text files with very simple formats. If the routine encountered a bookmark that was included in the bookmark collection of more than hundred participants, all tags and their distribution for the bookmark were additionally retrieved. Different test data was selected for each experiment. The test data suites are described in the following sections and available upon request.

4.2 Data selection

In the first experiment, we want to find a feasible method for selecting a subset of the provided data in which the principle of *interest-based locality* [12] can be

[1] The data was downloaded between June 21 and June 30, 2005.

Test set Nr.	1	2	3	4
Number of participants	551	155	248	280
Number of items	17575	5709	8861	10237
Number of unique items	12855	4311	6045	6643
in % of all items	73,14 %	75,51 %	68,22 %	64,89 %
Number of popular items	5691	2393	3691	4207
Number of unique popular items	2301	1217	1479	1483
in % of all unique items	17,90 %	28,23 %	24,47 %	22,23 %
Average number of items per user (Max: 50)	31,9	36,83	35,73	36,56
Average number of popular items per user	10,32	14,79	13,61	13,50

Table 1. Properties of the test sets

observed. This principle was originally observed in peer-to-peer environments. It means that if a participant A has a particular piece of content participant B is interested in, it is likely the case that the other information items stored by participant A are also of interest to participant B. As already discussed in Section 3, the user behaviour in a folksonomy is comparable to the user behaviour in peer-to-peer networks. Thus, we assume that interest-based locality can also be observed in folksonomies. Given the interfaces for data retrieval provided by *del.icio.us*, two different methods for data selection are possible:

- Select a certain tag and retrieve all bookmarks for this tag
- Select a certain bookmark and retrieve all participants that store this bookmark

Since we want to retrieve data from participants that form a community by sharing a common interest, and sharing the same bookmark is a stronger connection than sharing the same tag, we decided to choose the second option. First, a random bookmark b was chosen as a starting point. In the second step, the participant names of all participants that store b in their bookmark collection were retrieved. In the third step, the bookmark collections of all of these participants were downloaded. The fifty entries of each participant's bookmark collection which were added latest were included. For participants storing less than fifty entries, all existing entries were considered. Using the procedure described above, four test sets of different size were collected. The four random bookmarks[2] were chosen to be bookmarks that refer to Web sites containing information about diverse topics.

The properties of the test sets are described in Table 1. First of all, we can see that the total number of bookmarks of a test set is proportional to the number of participants that store bookmark b. Since the percentage of unique

[2] test set 1: b is `http://del.icio.us/url/06df5507a27ab5aa297fbb7748374df6`,
test set 2: b is `http://del.icio.us/url/463f3f6f9ce9471fef7f9edb881ad2d7`,
test set 3: b is `http://del.icio.us/url/245d7b2a49a80771da9a4d3a02d539c3`,
test set 4: b is `http://del.icio.us/url/c745432a483a84037c90e08d79f7c306`

All items shared ...	by > 10	by 5 - 10	by 4	by 3	by 2	not shared
Test set 1	0,41 %	1,84 %	1,34 %	2,76 %	9,09 %	84,56 %
Test set 2	0,16 %	1,83 %	1,37 %	2,85 %	9,21 %	84,57 %
Test set 3	0,58 %	2,45 %	1,70 %	2,96 %	8,68 %	83,62 %
Test set 4	0,90 %	2,60 %	1,55 %	3,10 %	8,63 %	83,22 %
Average	0,51 %	2,18 %	1,49 %	2,92 %	8,90 %	84,00 %
Popular items shared ...	by > 10	by 5 - 10	by 4	by 3	by 2	not shared
Test set 1	1,91 %	8,43 %	6,04 %	9,87 %	23,55 %	50,20 %
Test set 2	0,49 %	6,08 %	3,94 %	9,37 %	20,46 %	59,65 %
Test set 3	2,50 %	8,32 %	3,92 %	8,92 %	17,58 %	58,76 %
Test set 4	3,84 %	8,36 %	4,18 %	8,90 %	18,07 %	56,64 %
Average	2,19 %	7,80 %	4,52 %	9,27 %	19,92 %	56,30 %

Table 2. Distribution of bookmarks if (a) considering all bookmarks (top), or (b) considering popular bookmarks only (bottom)

bookmarks in each test set ranges from 64,89 % to 75,51 %, the number of unique bookmarks in a test set is not proportional to the total number of bookmarks. There are only small differences in the average number of bookmarks included in a participant's collection (Min: 31,9, Max: 36,83). The number of participants in a test set has a small impact on this number: In test set 1, which is by far the biggest of all sets, the average number of bookmarks per participant is higher than in the other test sets. Our decision to consider only the first fifty entries of each bookmark collection was based on the assumption that a high percentage of all *del.icio.us* participants stores more than fifty entries. We can observe from the retrieved data that this is not the case. On average, only 1,15 percent of participants in each set own a collection which is comprised of fifty bookmarks (and probably more that we did not retrieve). A bookmark is defined to be a popular bookmark if it is included in the bookmark collection of more than hundred *del.icio.us* participants. On average, a third of all bookmarks fall in the category of popular bookmarks. As can be seen in test set 1 and test set 2, the smaller the test set in terms of number of participants, the higher the amount of popular bookmarks per user.

Next, we analyse the distribution of bookmarks in the test sets. All unique bookmarks of a test set were considered. The results of this analysis are shown in Table 2. For each bookmark, we determined the total number of participants that store it in their collection. It turns out that the distributions of unique bookmarks share equal properties in each test set. On average, only 0,51 % of the bookmarks are stored by more than ten participants. 2,18 % are stored by a group of five to ten participants. 1,49 % are stored by four participants. 2,92 % are stored by three participants. 8,90 % are stored by 2 participants. The percentage of bookmarks that are not shared, but stored in only one participant's collection, is nearly equal in each test and on average 84 %. This is a very high value that shows that our data retrieval method as described above is not sufficient

for selecting subsets of the *del.icio.us* data which conform to the principle of interest-based locality. Hence, we want to know if interest-based locality can be observed when considering only the popular bookmarks. As can be seen in the bottom of Table 2, in this case the percentage of bookmarks that are present in only one collection lowers to 56,3 %. On average, 19,92 % percent of all popular bookmarks are shared by two participants, 9,27 % are shared by three participants, 4,52 % by four participants, 7,8 % are shared by between five and ten participants, and 2,19 % by more than ten participants.

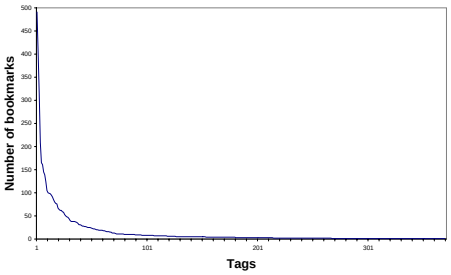

Fig. 3. Top tag distribution (ranked plot, linear scale) in set 1

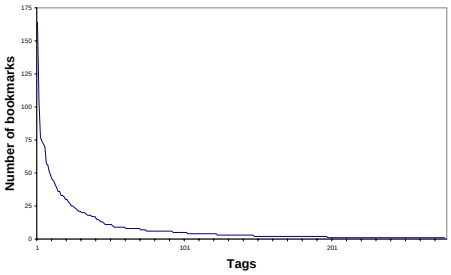

Fig. 4. Top tag distribution (ranked plot, linear scale) in set 2

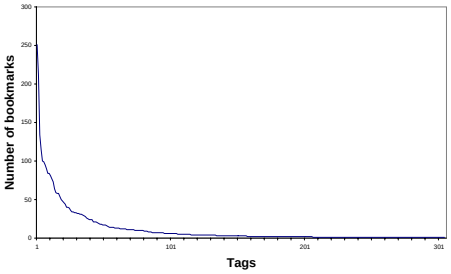

Fig. 5. Top tag distribution (ranked plot, linear scale)in set 3

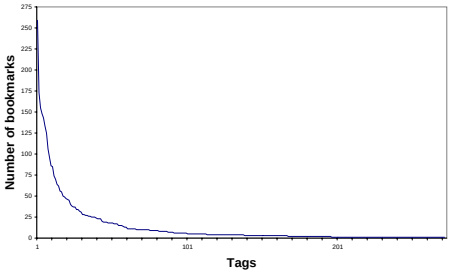

Fig. 6. Top tag distribution (ranked plot, linear scale) in set 4

Assuming that bookmarks that refer to Web sites with similar topics are annotated with the same tags, we now consider the tags associated with the popular bookmarks. For each bookmark, the top tag that was used by most participants was considered. The distributions of the top tags for all popular bookmarks are shown in Figure 3 to Figure 6. As can be seen from these figures, the distribution curves for all four test sets show equal properties. There is a long tail in each curve that reveals that there are many top tags that are included only once. The reason for that lies in the diversity of bookmark collections. Too many

Item present in ...	one set	two sets	Item present in ...	one set	two sets	three sets
1 and 2	92,28 %	7,72 %	1,2, and 3	86,27 %	10,12 %	3,61 %
1 and 3	89,71 %	10,29 %	1,2, and 4	86,96 %	9,53 %	3,51 %
1 and 4	90,55 %	9,45 %	1,3, and 3	84,32 %	11,58 %	4,10 %
2 and 3	87,47 %	12,53 %	2,3, and 4	82,03 %	12,50 %	5,48 %
2 and 4	87,88 %	12,12 %				
3 and 4	85,09 %	14,91 %				

Table 3. Bookmarks present in more than one test set when comparing (a) two test sets (left), or (b) three test sets (right)

of them contain items about topics that are not related to the topics of the other items of the collection. Hence, even when considering popular bookmarks only, it is not possible to retrieve test sets that conform to the principle of interest-based locality without any further preparation of the data. The necessary preparations consist of removing all items that cause the tail of the top tag distribution.

4.3 Joining data

In this analysis, we want to find out if it is possible to join test sets in order to create one bigger test set out of them. The question is if there are any connections between the data and to which extend the test sets are overlapping. Two kinds of overlaps are possible. The first is the overlap of participants, where one participant is present in more than one of the data subsets. Analysing the distribution of participants, the four test sets are nearly disjoint. The majority of participants (96.04 %) is included in only one test set. 3.96 % of participants were included in two sets. No participant was included in three or all four test sets. The second possibility for overlaps is that bookmarks can be present in more than one set. Table 3 shows the results of comparing the test sets to each other. When comparing two test sets, on average 11,17 % are included in both sets. When comparing three test sets, on average 4,18 % are in all three sets and 10,93 % in two sets. When joining all four sets, 82,33 % of bookmarks are present in one set, 11,72 % in two sets, 3,63 % in three, and 2,32 % are present in all four sets. Hence, it is not possible to use the method described in Section 4.2 to retrieve data and to join these sets to create bigger sets. The overlap between the sets is too small.

4.4 Data semantics

In this experiment, we compare the meta-data provided by the *del.icio.us* folksonomy to an already existing annotated bookmark collection built by the *DMOZ Project* [11]. This project is an effort of a community of volunteers to build a taxonomy for Web pages and to categorize Web pages according to this taxonomy. Since the DMOZ project that has already been used for simulating user

```
URL http://arxiv.org/
DMOZ Top/Science/Physics/Publications
DMOZ Top/Science/Math/Publications
DMOZ Top/Science/Math/Publications/Online_Texts/Collections
DMOZ Top/Science/Publications/Archives/Free_Access_Online_Archives
ID 19aa8ff1e9e2a06677ab34f3f2a5b0c8
TITLE arXiv.org e-Print archive
TAGS physics:43;science:41;research:27;math:23;papers:19;reference:18;ma
thematics:15;journal:10;articles:10;archive:9;biology:8;eprint:7;library
:7;preprint:6;books:6;programming:6;cs:5;article:5;academic:5;computer:4
;arxiv:4;literature:4;toread:4;computerscience:4;ai:3;study:3;
```

Fig. 7. A sample entry containing meta-data from both sources

behaviour in a peer-to-peer network [13], it is interesting for us to know to which extend the meta-data provided by the DMOZ project is similar to the meta-data provided by *del.icio.us*. For conducting this experiment, we downloaded the RDF dump[3] of the structure and of the contents of the DMOZ directory and stored it in a relational database for performance reasons. After that, a database lookup for each popular bookmark included in the test sets described in Section 4.2 was performed to check if the bookmark is included in the DMOZ contents as well. Each time the lookup routine encountered a hit, the bookmark and its meta-data from both sources were appended to a text file with a very simple format (see Figure 7 for an example). If a bookmark was assigned more than one DMOZ topic, we considered all of them. Two observations we made while performing this task are worth mentioning:

- The intersection of *del.icio.us* and the DMOZ directory is rather small. Although only popular bookmarks were used, only 25 % of the bookmarks were also included in the contents of the DMOZ directory.
- Nearly 50 % of those bookmarks that are present both in both sources are instances of subtopics of the DMOZ topic `Top/Computers`.

In total, the test data for this experiment consists of 788 bookmarks together with all corresponding DMOZ topics and all tags and their numbers from *del.icio.us*. All DMOZ topics were considered except of the subtopics of topic `Top/World`, which is the branch in the DMOZ hierarchy that builds the top concept for multi-lingual categories not defined in the English language. All DMOZ topic names were converted to lower-case characters. Underscores and hyphens were removed from both topic and tag names. To overcome the problem that singular and plural versions of tags are in use, in case the last character of a tag or topic name was the letter `s`, it was removed (e.g., `computers` was changed to `computer`). Some DMOZ topics have 26 subtopics for each letter from A to Z in order to categorize items by the first letter of their name. Such topics consisting of only one character were removed from the topic path. The topic `Top` was removed from each topic path. Since the leaf entry from the DMOZ topic path is the one that most exactly categorizes a bookmark, we sorted the topic paths to

[3] available at `http://rdf.dmoz.org`

	1st	2nd	3rd	4th	5th	6th	7th to 11th
Top tag	9,44 %	15,94 %	12,67 %	4,72 %	3,28 %	1,72 %	0,81 %
Top 3 tags	20,37 %	27,55 %	21,58 %	14,29 %	12,23 %	6,21 %	2,30 %
Top 5 tags	28,32 %	34,81 %	27,72 %	19,75 %	16,42 %	11,03 %	3,69 %
Top 10 tags	37,38 %	44,53 %	35,94 %	27,08 %	25,91 %	18,28 %	6,25 %
Top 15 tags	44,30 %	52,45 %	43,17 %	34,16 %	32,12 %	26,55 %	8,93 %
All tags	52,99 %	62,55 %	52,48 %	46,34 %	44,34 %	40,34 %	14,73 %

Table 4. Comparison of categorization data from both sources, considering 1, 3, 5, 10, 15, or all tags for a given bookmark.

their reverse order. For example, the topic path shown in Figure 7 is converted to `publications physics science`.

On average, the topic path length of all DMOZ topics prepared as described above is 4,67. The average number of tags per bookmark is 24,59. The following method is employed for comparing topics to tags. Topics are used as a reference and tags are compared to them. If more than one topic is assigned to a bookmark, a separate comparison for each topic is performed. One comparison consists of several lookups, one for each entry of a topic path, e.g., for `publications physics science` three lookups are performed. The result of a lookup is either true in case of a match, or false. The results of this comparison are shown in Table 4. It turns out that the leaf entries of the topic path match more often than the top entries. This is not surprising, since the top entries are very general, e.g., `Computers`, `Arts`, or `Science` and only a few *del.icio.us* participants will use general terms to describe their items. When taking into account only the top tag, which is the one that most participants used for annotating, the highest percentage of matches is 15,94 % for the second entry of each topic path. It can be seen that the values rise linearly. The more tags are taken into account, the better the results. The fairest comparison is that of the top five tags, since this is the average number of the topic path length. In this case, the highest percentage of matches is 34,81 % for the second entry of each topic path.

In summary, it can be seen that the terms used for categorization are very different in both sources. Even when comparing all tags to each topic path entry and hence conducting on average 24 comparisons of tags to one single topic, there is no match in 37,45 % to 85,27 % of the cases.

4.5 Popularity of items

In the last experiment, our assumption is that users are more interested in a certain bookmark if it is already included in many other bookmark collections than if it is included in only a few. Hence, bookmarks that are already popular will become even more popular. In particular, we want to know if the list of popular bookmarks[4] which shows those bookmarks that most users added to

[4] available at `http://del.icio.us/popular/`

their collections recently has an impact on the popularity of a bookmark. We observed this list several times for the time span of a day and collected a snapshot every 10 minutes. These snapshots are comprised all listed bookmark's URLs and the total number of persons that included it in their bookmark collection at the given point in time. For clarity, Figure 8 and Figure 9 show only those bookmarks that were present in the list of popular bookmarks for the complete time of observation. Analysing these data, one can see in Figure 8 that the assumption is true to some extend, since those curves that are higher increase a little faster than those that are low, but the differences are not significant as we expected.

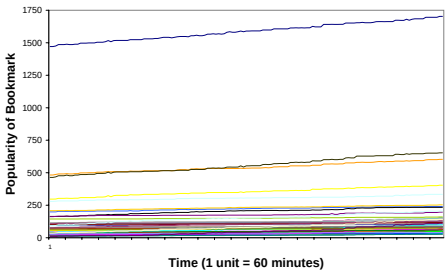

Fig. 8. Popular bookmarks on Tuesday, 28th of June

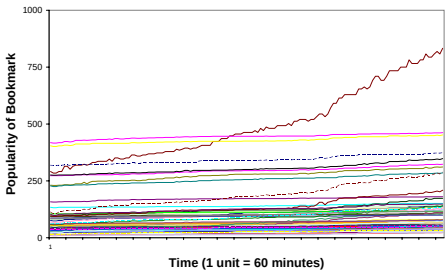

Fig. 9. Popular bookmarks on Thursday, 23th of June

Figure 9 shows that there are other factors that have more impact on the popularity of bookmarks over time than being included in the list of popular bookmarks. There is one particular bookmark with its popularity rising very quickly while all other bookmarks show the same performance as in Figure 8. Hence, the reason for this significant increase is not caused by being included the list of popular bookmarks.

5 Conclusion

In this paper, a method for selecting subsets of the meta-data provided by a folksonomy that adhere to the principle of interest-based locality was developed. The resulting data can be applied for simulating peers and their contents in a peer-to-peer environment. The properties of the test sets that were retrieved by using this method were analysed and discussed in order to prove that the proposed method selects subsets that have similar properties. Comparing the meta-data produced by the folksonomy to meta-data created by the DMOZ open directory project at the data level revealed that there are major differences between them. Finally, we showed that centrally provided lists of popular items have only small influences on the properties of these items.

Acknowledgements

We thank Marion Murzek and Gerhard Kramler for helpful comments on earlier versions of this paper.

References

1. K. Aberer, P. Cudre-Mauroux, and M. Hauswirth. The Chatty Web: Emergent Semantics Through Gossiping. In *Procceedings of the 12th International World Wide Web Conference (WWW 2003)*, May 2003.
2. K. Aberer, P. Cudre-Mauroux, A. M. Ouksel, T. Catarci, M.-S. Hacid, A. Illarramendi, V. Kashyap, M. Mecella, E. Mena, E. J. Neuhold, O. D. Troyer, T. Risse, M. Scannapieco, F. Saltor, L. de Santis, S. Spaccapietra, S. Staab, and R. Studer. Emergent Semantics Principles and Issues. In *Procceedings of the 9th International Conference on Database Systems for Advanced Applications (DASFAA 2004)*, March 2004.
3. Adam Mathes. Folksonomies – Cooperative Classification and Communication Through Shared Metadata. `http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html`, December 2004.
4. M. Bernauer, G. Kappel, and E. Michlmayr. Traceable Document Flows. In *Proceedings of the 2nd International Workshop on Web Semantics (WebS), DEXA2004*, September 2004.
5. B. Chandrasekaran, J. R. Josephson, and R. Benjamins. What are Ontologies, and Why Do We Need Them? *IEEE Intelligent Systems and Their Applications*, 14(1):20–26, 1999.
6. Clay Shirky. Ontology is Overrated: Categories, Links, and Tags. `http://shirky.com/writings/ontology_overrated.html`, 2005.
7. T. Hammond, T. Hannay, B. Lund, and J. Scott. Social Bookmarking Tools (I): A General Review. *D-Lib Magazine*, 11(5), April 2005.
8. Joshua Schachter. Del.icio.us Social Bookmarking Service. `http://del.icio.us/`, 2005.
9. B. Lund, T. Hammond, M. Flack, and T. Hannay. Social Bookmarking Tools (II): A Case Study – Connotea. *D-Lib Magazine*, 11(4), April 2005.
10. E. Michlmayr, S. Graf, W. Siberski, and W. Nejdl. Query Routing with Ants. In *Proceedings of the 1st Workshop on Ontologies in P2P Communities, ESWC2005*, May 2005.
11. Netscape Communications Corporation. DMOZ Open Directory Project. `http://www.dmoz.org`, 2005.
12. K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems. In *Proceedings of IEEE INFOCOM 2003*, April 2003.
13. C. Tempich, S. Staab, and A. Wranik. REMINDIN': Semantic Query Routing in Peer-to-Peer Networks based on Social Metaphors. In *Proceedings of the 13nd International World Wide Web Conference (WWW2004)*, May 2004.
14. Yahoo! Company. Flickr Online Photo Management Service. `http://flickr.com/`, 2005.

Personalizing Applications through Integration of Inferred Trust Values in Semantic Web-Based Social Networks

Jennifer Golbeck
Department of Computer Science
University of Maryland, College Park
College Park, Maryland, USA
golbeck@cs.umd.edu

Abstract. Social Network data, represented using the FOAF Vocabulary, is some of the most prevalent data on the Semantic Web. In this work, we look particularly at trust relationships in web-based social networks and their implications for software personalization. We present a network analysis as the foundation for TidalTrust, and algorithm for inferring trust relationships, and then illustrate how the results can be used to improve application interfaces.

1 Introduction and Background

The Friend Of A Friend (FOAF) project is one of the most popular efforts on the Semantic Web. The vocabulary for describing people and their social network connections is already used to represent information about over 8,000,000 people, and is a form of output being used by many large web-based social networks. This huge source of distributed data offers opportunities for performing social network analysis on real, evolving networks, and the web-based nature means that the publicly accessible data can be computed against and integrated into applications to help benefit the user.

In addition to the core vocabulary, FOAF has been extended in several ways to enhance the information about interpersonal relationships. For example, the FOAF Relationship Module[1] offers dozens of relationship types, such as "sibling of", "would like to know", and "spouse of". This work utilizes the FOAF Trust Module[2] which allows users to indicate how much they trust people they know.

With a social network where users have indicated how much they trust others, it is possible to recommend (or infer) how much one user might trust an unknown person by using the trust values on the paths that connect them. By inferring the trustworthiness of an unknown person, the quality of information from that person can be judged. In this paper, we present an algorithm, Tidal Trust, for inferring trust relationships in Semantic Web-based social networks. We describe the social network analysis that leads to these algorithms, and describe their accuracy within two networks. The benefit of this analysis is that the results can be integrated into applications to enhance the users' experiences by acting with respect to their social preferences. We present two applications – FilmTrust and TrustMail – that utilize the Tidal Trust algorithm, and show how the trust information leads to usability enhancements.

[1] http://www.perceive.net/schemas/20021119/relationship/

1.1 Experimental Networks

One facet of web-based social networks that makes them more complex and interesting that networks traditionally studied with social network analysis is the plethora of information about relationships that is available. In this work, we were particularly interested in trust because of the potential that information has for improving applications. Recent work provides two indications that users will prefer the sort of system that relies on trust in social networks. First, users tend to prefer recommendations from people they know and trust [3]. Related work also showed that users prefer recommendations from systems that they trust and understand [4]. Ziegler and Lausen [5] also showed a correlation between trust and user similarity in an empirical study of a real online community, indicating that trusted people may direct users to data relevant to their interests.

To perform the analysis, it was necessary to have data sources. Two separate trust networks have been grown from scratch. The first network is part of the Trust Project at http://trust.mindswap.org/. This network is built up from distributed data maintained on the Semantic Web. Within their FOAF files, users include trust ratings for people they know using the FOAF Trust Module, a simple ontology for expressing trust developed as part of this project. The ontology has vocabulary for rating people on a scale of 1 (low trust) to 10 (high trust). These ratings can be made in general or with respect to a specific topic. In the network built up for study in this research, users assigned general ratings to one another. There are approximately 2,000 people in this network with over 2,500 connections. Figure 1 shows the current structure of this network.

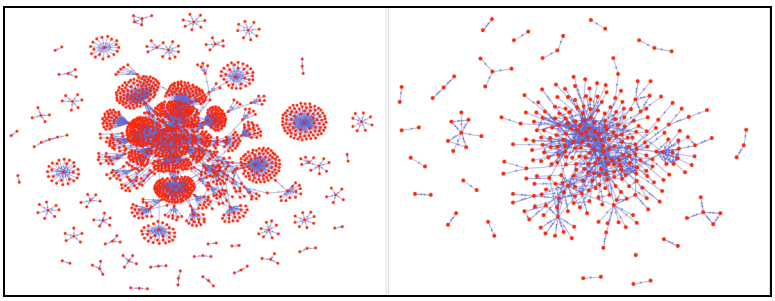

Fig 1. The structure of the social network from the Trust Project (left) and FilmTrust (right).

The second network is part of the FilmTrust project, a website that combines social networks with a movie ratings and reviews site. The site currently comprises 500 members who have rated each others' trustworthiness on the same 1-10 scale. In this network, users rate how much they trust people about movies.

1.2 Related Work

The issue of sharing trust assessments on the semantic web has been addressed in contexts outside of explicit social networks. Gil and Ratnakar addressed the issue of trusting content and information sources [6] on the Semantic Web. Their TRELLIS system derives assessments about information sources based on individual

feedback about the sources. Our work uses this notion of augmenting data on the Semantic Web (social network data in our case) with annotations about its trustworthiness.

Once a trust network has been properly modeled and represented, our attention moves to algorithms for calculating recommendations about trust in the network. The question of trust calculations in social networks has been addressed in several communities with a range of endpoint applications.

The EigenTrust algorithm [7] is used in peer-to-peer systems and calculates trust with a variation on the PageRank algorithm [8], used by Google for rating the relevance of web pages to a search. EigenTrust is designed for a peer-to-peer system while ours is designed for use in humans' social networks, and thus there are differences in the approaches to analyzing trust. In the EigenTrust formulation, trust is a measure of performance, and one would not expect a single peer's performance to differ much from one peer to another. Socially, though, two individuals can have dramatically different opinions about the trustworthiness of the same person. Our algorithms intentionally avoid using a global trust value for each individual to preserve the personal aspects that are foundations of social trust.

Raph Levin's Advogato project [9] also calculates a global reputation for individuals in the network, but from the perspective of designated *seeds* (authoritative nodes). His metric composes certifications between members to determine the trust level of a person, and thus their membership within a group. While the perspective used for making trust calculations is still global in the Advogato algorithm, it is much closer to the methods used in this research. Instead of using a set of global seeds, we let any individual be the starting point for calculations, so each calculated trust rating is given with respect to that person's view of the network.

Richardson et. al.[10] use social networks with trust to calculate the belief a user may have in a statement. This is done by finding paths (either through enumeration or probabilistic methods) from the source to any node which represents an opinion of the statement in question, concatenating trust values along the paths to come up with the recommended belief in the statement for that path, and aggregating those values to come up with a final trust value for the statement. Current social network systems on the Web, however, primarily focus on trust values between one user to another, and thus their aggregation function is not applicable in these systems.

2 Inferring Trust in Social Networks: TidalTrust

Inferring trust relationships within a social network requires an analysis of the properties of trust networks. We start with the assumption that people who are trusted highly will tend to agree with the user more about the trustworthiness of others than people who are less trusted. In this section, we present an analysis of the features of a trust network and their correlation to accuracy. Those results are then used in the development of an algorithm, TidalTrust, for inferring trust relationships

2.1 Correlation of Trust and Accuracy

To investigate the correlation of trust and accuracy, experiments were performed on the Trust Project network. The goal was to ascertain if neighbors with higher trust ratings were more likely to agree with the source about the trustworthiness of a third

person. This was determined repeating the following process for each node. First, a node was chosen as the source. For each neighbor of the source, n_i, a list of common neighbors of the source and n_i was compiled. For each of those common neighbors, the difference between the source's rating and n_i's rating was recorded as a measure of accuracy. A smaller difference means a higher accuracy. This difference was recorded along with the source's rating of n_i. Figure 2 illustrates one step in the process.

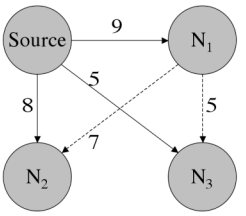

Fig. 2. Finding points of comparison in the network. In these experiments, this network would produce two data points: the difference between the source and N_1's ratings of N_2 (in this case, 1) and the difference between the source and N_1's ratings of N_3 (in this case, 0)

These experiments produced a pair of numbers for each data point: the trust value from the source to its neighbor (n_1), and the difference between the ratings of a common neighbor (Δ). The number of data points for each trust value indicates how frequently common neighbors are shared between pairs of nodes at each trust level.

The frequency of common neighbors among pairs of nodes with high trust levels is much higher than the frequency of those ratings in the original network. These indicate that people with stronger trust connections share more common social connections. In fact, over 40% of the common neighbors were found between nodes that shared a high trust rating. If the experimental results show that the results are more *accurate* when there is more trust, this distribution means that a the increased accuracy will be reinforced by the increased frequency of common neighbors among pairs with high trust.

If individuals with higher trust ratings agree with the source more, we would expect average difference (Δ) would decrease as trust ratings increase. In the datasets used, there were very few comparisons available for trust ratings 1-5. Because the number of comparisons for the lower trust ratings is so small, and the margin of error so large, these data points were not included in the analysis here. Instead, we focused on the comparisons made for trust values 6-10.

As shown in Figure 3, there appears to be a strong negative linear relationship between trust value and Δ. This is confirmed by the statistics; the Pearson's correlation is –0.991, indicating that there is an almost perfect negative linear relationship between the variables. These results are statistically significant for $p<.001$.

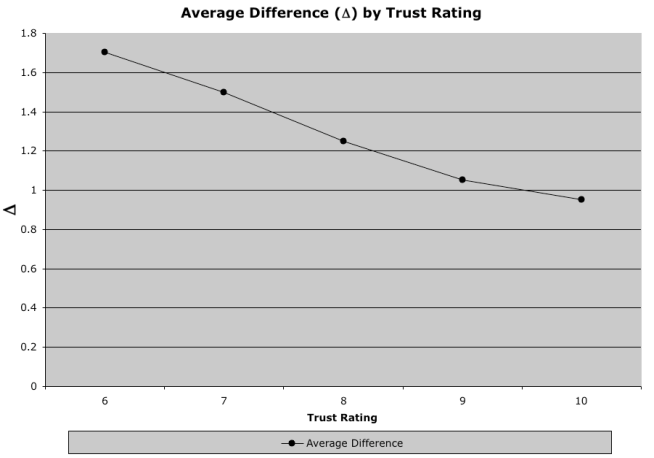

Fig 3. The relationship between Δ and Trust Rating.

These analyses show that in this trust network, there is more agreement between nodes connected by high trust ratings than nodes connected by lower trust ratings. Furthermore, common neighbors are found more frequently among pairs of nodes with higher trust ratings. Thus, the increased accuracy among highly trusted neighbors is amplified by the increased frequency of receiving data from those highly trusted neighbors. This leads to results that are more accurate overall. These elements will become a critical in the development of the trust inference algorithm.

2.2 Path Length and Accuracy

The length of a path is determined by the number of edges the source must traverse before reaching the sink. For example, source→n_1→sink has length two. Does the length of a path affect the agreement between individuals? Specifically, should the source expect that neighbors who are connected more closely will give more accurate information than people who are further away in the network? To study this relationship between path length and accuracy, we follow a similar approach used in section 2.1. The source is selected and for each source's neighbor n_i, we search for paths of length l to n_i. We compare the source's rating of n_i to the rating given to n_i along the path of length l.

Table I. Minimum average Δ for paths of various lengths containing the specified trust rating.

Trust Rating \ *Path Length*	2	3	4	5
10	0.953	1.52	1.92	2.44
9	1.054	1.588	1.969	2.51
8	1.251	1.698	2.048	2.52
7	1.5	1.958	2.287	2.79
6	1.702	2.076	2.369	2.92

Figure 4 illustrates the relationships from Table I.

For each path length, Table I shows the minimum average Δ. These are grouped according to the minimum trust value along that path.

In Figure 4, the effect of path length can be compared to the effects of trust ratings. For example, consider the average Δ for trust values of 7 on paths of length 2. This is approximately the same as the average Δ for trust values of 10 on paths of length 3 (both are close to 1.5). The average Δ for trust values of 7 on paths of length 3 is about the same as the average Δ for trust values of 9 on paths of length 4. A precise rule cannot be derived from these values because there is not a perfect linear relationship, and also because the points in Figure 4 are only the minimum average Δ among paths with the given trust rating.

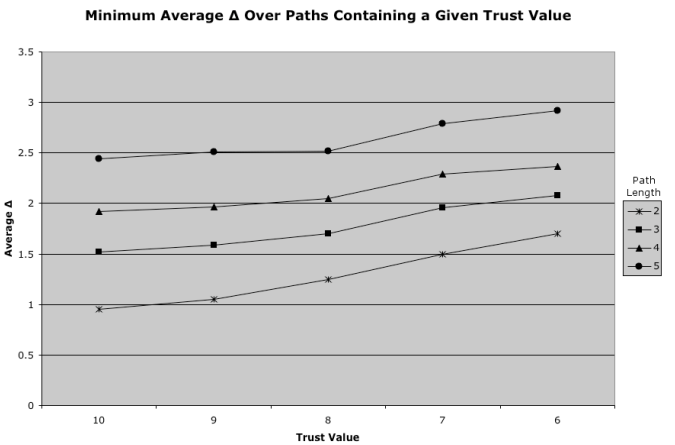

Fig 4. Minimum average Δ from all paths of a fixed length containing a given trust value.

This relationship will be integrated into the algorithms for inferring trust presented in the next section.

2.3 TidalTrust: An Algorithm for Inferring Trust

The in-depth look at the effects of trust ratings and path length in the previous section guided the development of TidalTrust, an algorithm for inferring trust in networks with continuous rating systems. The following guidelines can be extracted from the analysis of the previous sections:

1. For a fixed trust rating, shorter paths have a lower average Δ.
2. For a fixed path length, higher trust ratings have a lower average Δ.

This section describes how these features are used in the TidalTrust algorithm.

2.3.1 Incorporating Path Length The analysis in section 2.2 indicates that a limit on the depth of the search should lead to more accurate results, since the average Δ increases as depth increases. If accuracy decreases as path length increases, as the earlier analysis suggests, then shorter paths are more desirable. However, the tradeoff is that fewer nodes will be reachable if a limit is imposed on the path depth. To

balance these factors, the path length can vary from one computation to another. Instead of a fixed depth, the shortest path length required to connect the source to the sink becomes the depth. This preserves the benefits of a shorter path length without limiting the number of inferences that can be made.

2.3.2 Incorporating Continuous Trust The previous results also indicate that the most accurate information will come from the highest trusted neighbors. As such, we may want the algorithm to limit the information it receives so that it comes from only the most trusted neighbors, essentially giving no weight to the information from neighbors with low trust. If the algorithm were to take information only from neighbors with the highest trusted neighbor, each node would look at its neighbors, select those with the highest trust rating, and average their results. However, since different nodes will have different maximum values, some may restrict themselves to returning information only from neighbors rated 10, while others may have a maximum assigned value of 6 and be returning information from neighbors with that lower rating. Since this mixes in various levels of trust, it is not an ideal approach. On the other end of possibilities, the source may find the maximum value it has assigned, and limit every node to returning information only from nodes with that rating or higher. However, if the source has assigned a high maximum rating, it is often the case that there is no path with that high rating to the sink. The inferences that are made may be quite accurate, but the number of cases where no inference is made will increase. To address this problem, we define a variable *max* that represents the largest trust value that can be used as a minimum threshold such that a path can be found from source to sink.

$$t_{is} = \frac{\sum_{j \in adj(i) \ni t_{ij} >= \max} t_{ij} t_{js}}{\sum_{j \in adj(i) \ni t_{ij} >= \max} t_{ij}} \qquad (1)$$

2.3.3 Full Algorithm for Inferring Trust Incorporating the elements presented in the previous sections, the final TidalTrust algorithm can be assembled. The name was chosen because calculations sweep forward from source to sink in the network, and then pull back from the sink to return the final value to the source.

The source node begins a search for the sink. It will poll each of its neighbors to obtain their rating of the sink. Each neighbor repeats this process, keeping track of the current depth from the source. Each node will also keep track of the strength of the path to it. Nodes adjacent to the source will record the source's rating assigned to them. Each of those nodes will poll their neighbors. The strength of the path to each neighbor is the minimum of the source's rating of the node and the node's rating of its neighbor. The neighbor records the maximum strength path leading to it. Once a path is found from the source to the sink, the depth is set at the maximum depth allowable. Since the search is proceeding in a Breadth First Search fashion, the first path found will be at the minimum depth. The search will continue to find any other paths at the minimum depth. Once this search is complete, the trust threshold (the variable *max* in formula 1) is established by taking the maximum of the trust paths leading to the sink. This is illustrated in Figure 5 .

With the *max* value established, each node can complete the calculations of a weighted average by taking information from nodes that they have rated at or above the *max* threshold.

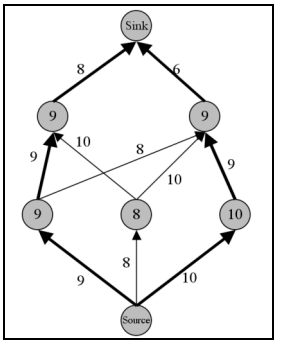

Fig 5. The process of determining the trust threshold. The label on each edge represents the trust rating between nodes. The label on each node indicates the maximum trust strength on the path leading to that node. The two nodes adjacent to the sink have values of 9, so 9 is the *max* value. The bold edges indicate which paths will ultimately be used in the calculation because they are at or above the *max* threshold.

2.4 Accuracy of TidalTrust

As presented above, TidalTrust strictly adheres to the observed characteristics of trust: shorter paths and higher trust values lead to better accuracy. However, there are some things that should be kept in mind. The most important is that networks are different. Depending on the subject (or lack thereof) about which trust is being expressed, the user community, and the design of the network, the effect of these properties of trust can vary. While we should still expect the general principles to be the same – shorter paths will be better than longer ones, and higher trusted people will agree with us more than less trusted people – the proportions of those relationships may differ from what was observed in the sample networks used in this research.

Table II. Average Δ for TidalTrust and Simple Average recommendations in both the Trust Project and FilmTrust networks. Numbers are absolute error on a 1-10 scale.

	Algorithm	
Network	**TidalTrust**	**Simple Average**
Trust Project	1.09	1.43
FilmTrust	1.35	1.93

There are several algorithms that output trust inferences, but none of them produce values within the same scale that users assign ratings. Some trust algorithms form the Public Key Infrastructure (PKI) are more appropriate for comparison. A comparison of this algorithm to PKI can be found in [2], but due to space limitations that comparison is not included here. One direct comparison to make is to compare the average Δ from TidalTrust to the average Δ from taking the simple average of all

ratings assigned to the sink as the recommendation. As shown in Table II, the TidalTrust recommendations outperform the simple average in both networks, and these results are statistically significant with $p<0.01$. Even with these preliminary promising results, TidalTrust is not designed to be the optimal trust inference algorithm for every network in the state it is presented here. Rather, the algorithm presented here adheres to the observed rules of trust. When implementing this algorithm on a network, modifications *should be made* to the conditions of the algorithm that adjust the maximum depth of the search, or the trust threshold at which nodes are no longer considered. How and when to make those adjustments will depend on the specific features of a given network. These tweaks will not affect the complexity of implementation.

3 Applying the Analysis: FilmTrust and TrustMail

In this section, we look at using trust values as recommender systems. Similar techniques for integrating trust and recommendations have appeared in recent work, including that by Massa et al [11], and Zeigler [5].

3.1 FilmTrust: Social Networks and Movie Ratings

FilmTrust is a website that utilizes trust ratings in a social network to make personalized predictive recommendations about movies to the user. Using the trust inferences from TidalTrust in an algorithm for generating ratings, we are able to show that in certain cases, the predictive ratings are far more accurate than more traditional methods.

The social networking component of the website requires users to provide a trust rating for each person they add as a friend. When creating a trust rating, users are advised to rate how much they trust their friend about movies. The other features of the website are movie ratings and reviews. Users can choose any film and rate it on a scale of a half star to four stars. They can also write free-text reviews about movies.

The social network is integrated with the movie ratings with the "Recommended Rating" feature. This is personalized using the trust values for the people who have rated the film (the *raters*). The process for calculating this rating is done with a weighted average, similar to the process for calculating trust ratings. Using the Tidal Trust algorithm, set of highly trusted nodes who have rated the given film are chosen. For the set of selected nodes S, the recommended rating r from node s to movie m is computed as the average of the movie ratings from nodes in S weighted by the trust value t from s to each node:

$$r_{sm} = \frac{\sum_{i \in S} t_{si} r_{im}}{\sum_{i \in S} t_{si}}$$

This average is rounded to the nearest half-star, and that value becomes the "Recommended Rating" that is personalized for each user.

As a simple example, consider the following:

- Alice trusts Bob 9
- Alice trusts Chuck 3
- Bob rates the movie "Jaws" with 4 stars

- Chuck rates the movie "Jaws" with 2 stars

Then Alice's recommended rating for "Jaws" is calculated as follows:

$$\frac{t_{Alice->Bob} * r_{Bob->Jaws} + t_{Alice->Chuck} r_{Chuck-Jaws}}{t_{Alice->Bob} + t_{Alice->Chuck}} = \frac{9*4+3*2}{9+3} = \frac{42}{12} = 3.5$$

Judging the accuracy of these ratings can also be done in a way similar to the analysis of the accuracy of the trust calculations. For each movie the user has rated, the recommended rating can be compared to the actual rating that the user assigned. For further comparison, we also compared the user's rating to the simple average rating of a movie (commonly shown on other movie websites), and to a recommended rating generated by a Pearson correlation-based automated collaborative filtering (ACF) algorithm [12].

On first analysis, it did not appear that that the personalized ratings offered any benefit over the average. The difference between the actual rating and the trust-based *recommended* rating (call this ∂r) was not statistically different than the difference between the actual rating and the *average* rating (call this ∂a). A close look at the data suggested why. Most of the time, the majority of users actual ratings are close to the average. Of course, it should be expected that there is a relatively normal distribution of ratings around the mean, and that a large percentage of ratings will fall close to that mean. A random sampling of movies showed that about 50% of all ratings were within +/- a half star of the mean. For these users, a personalized rating could not offer much benefit over the average. However, the point of the recommended rating is to perform well when the average does not, i.e. when the user's opinion is different from the average opinion or ∂a is high. In those cases, the personalized rating should give the user a better recommendation, because we expect the people they trust will have tastes similar to their own [5].

To test if this benefit is real, and experiment was conducted by computing the ∂ values with a minimum threshold on ∂a. The first set of comparisons was taken with no threshold, where the difference between ∂a and ∂r was not significant. As the minimum ∂a value was raised, a smaller group of user-film pairs were selected where the users made ratings that differed increasingly with the average. We incremented the minimum threshold of ∂a by 0.5, and then computed the new ∂ values The results are shown in Figure 6.

Notice that the ∂a value increases about as expected. The ∂r, however, is clearly increasing at a slower rate than ∂a. At each step, as the threshold for ∂a is increased by 0.5, ∂r increases by an average of less than 0.1. A two-tailed t-test shows that at each step where the minimum ∂a threshold is greater than or equal to 0.5, the recommended rating is significantly closer to the actual rating than the average rating is, with $p<0.01$. When compared to the ACF algorithm, there were similar results. For $\partial a<1$, there was no significant difference between the accuracy of the ACF ratings and the trust-based recommended rating. However, when the gap between the actual rating and the average increases, for $\partial a>=1$, the trust-based recommendation outperforms the ACF as well as the average, with $p<0.01$.

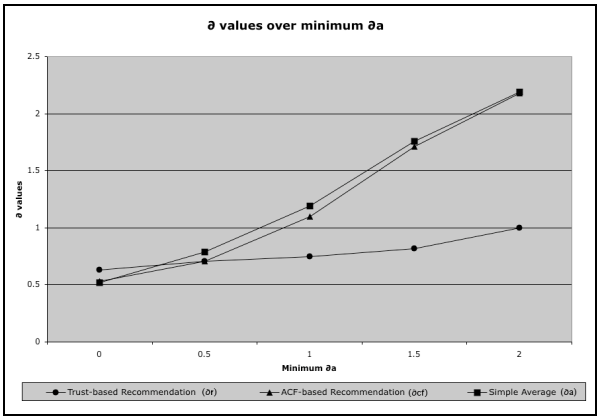

Fig 6. The increase in ∂ as the minimum ∂a is increased. Notice that the ACF-based recommendation (∂cf) follows the average (∂a). The more accurate Trust-based recommendation (∂r) significantly outperforms both other methods.

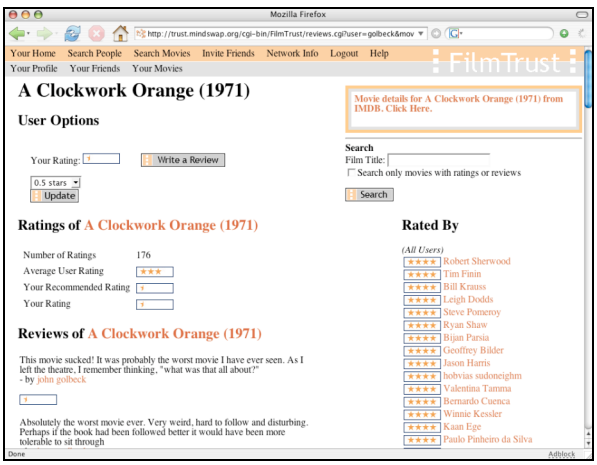

Fig 7. A user's view of the page for "A Clockwork Orange," where the recommended rating matches the user's rating, even though ∂a is very high (∂a = 2.5).

Figure 7 shows a clear example of the personalized rating at work. "A Clockwork Orange" is one of the films in the database that has a strong collective of users who hated the movie, even though the average rating was 3 stars and many users gave it a full 4-star rating. For the user shown, the average rating of 3 stars is 2.5 stars above the user's actual rating while the recommended rating exactly matches the user's low rating of 0.5 stars. These are precisely the type of cases that the recommended rating is designed to address.

The purpose of this work is not necessarily to replace more traditional methods of collaborative filtering. It is very possible that a combined approach of trust with correlation weighting or another form of collaborative filtering may offer equal or better accuracy, and it will certainly allow for higher coverage. However,

these results clearly show that, in the FilmTrust network, basing recommendations on the expressed trust for other people in the network offers significant benefits for accuracy.

In addition to presenting personalized ratings, the experience of reading reviews is also personalized. The reviews are presented in order of the trust value of the author, with the reviews from the most trustworthy people appearing at the top, and those from the least trustworthy at the bottom. The expectation is that the most relevant reviews will come from more trusted users, and thus they will be shown first. A preliminary user study suggests that this ordering is beneficial to users, but further work is necessary to refine and confirm these results.

3.2 TrustMail: Trust Networks for Email Filtering

TrustMail is a prototype email client that adds trust ratings to the folder views of a message. This allows a user to see their trust rating for each individual, and sort messages accordingly. This is, essentially, a message scoring system. The benefit to users is that relevant and potentially important messages can be highlighted, even if the user does not know the sender. The determination of whether or not a message is significant is made using the user's own perspective on the trust network, and thus scores will be personalized to and under the control of each user.

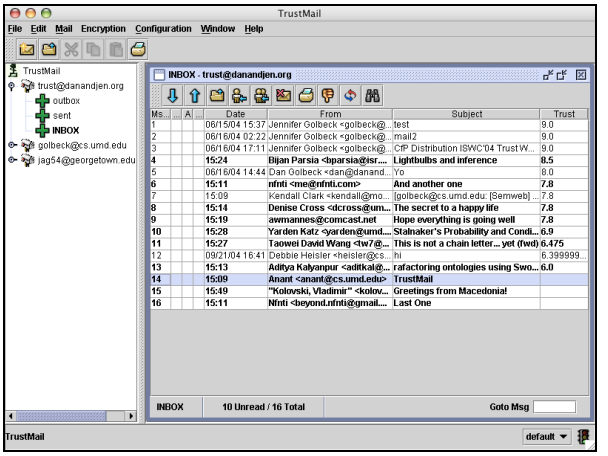

Fig 8.The TrustMail Interface. In this window, messages are sorted according to the trust rating of the sender, with the most trusted appearing highest in the list.

The values shown next to each message are trust ratings calculated with the TidalTrust algorithm were recipient as the source, and the sender is the sink. Techniques that build social networks from messages that the user has sent or received can identify whether or not a message has come from someone in the network. However, because they are built only from the user's local mail folders, no information is available about people that the user has not previously seen. If the user's personal network is connected in to a larger social network with information from many other users, much more data is available. Previously unseen senders can be identified as part of the network.

Furthermore, since trust values are available in the system, the methods for inferring trust can be applied to present more information to the user about the sender of a message. In the FilmTrust system, preliminary studies suggest that users benefited from having movie reviews sorted by the trustworthiness of the author. These results also suggest a benefit from sorting messages by the trustworthiness of the sender in TrustMail. However, unlike the FilmTrust where every review was authored by someone in the social network, people will undoubtedly receive many email messages from people who are not in their social network. To understand what benefit TrustMail might offer to users, it is important to understand what percentage of messages we can expect to have ratings for in TrustMail. The next section uses a real email corpus to gain some insight into this question.

To gain some insight into how TrustMail may impact a user's mailbox, a large network with many users is required. The Enron email dataset is a collection of the mail folders of 150 Enron employees, and it contains over 1.5 million messages, both sent and received. There are over 6,000 unique senders in the corpus, and over 28,000 unique recipients. These numbers are much greater than the number of users whose mailboxes have been collected because they represent everyone who has sent a message to the users, everyone who has been cc-ed on a message to the users, and everyone the users have emailed. The collection was made available by the Federal Energy Regulatory Commission in late 2003 in response to the legal investigation of the company. Because the messages represent a single community, they are ideal for analyzing the potential of TrustMail. Each message in the corpus was read, and an edge was added from the sender to each of the recipients. This produced an initial social network, although the connections are weak. To be more sure that the links between people represented a relationship, connections were removed for any interactions that occurred only once; edges were only added from source to sink when the source had emailed the sink at least twice.

An analysis of the Enron network showed the following statistics:

- 37% of recipients had direct connections to people who sent them email in the social network; in other words, 37% of the time the recipient had emailed the sender of a received message.
- 55% of senders who were not directly connected to the recipient could be reached through paths in the social network.
- Thus, a total of 92% of all senders can be rated if trust values were present in the social network.

These numbers indicate that for users in a community like Enron, an application like TrustMail can provide information about a majority of the incoming messages. While the Enron corpus is a close community of users, it is reasonable to expect that, if users are properly supported in making trust ratings as part of their email client, a similarly high percentage of senders and messages would receive ratings in other contexts.

4 Conclusions

In this paper, we have used an analysis of the properties of trust networks to develop the TidalTrust algorithm for inferring trust relationships between people with no direct connections. We integrated this algorithm into two systems: FilmTrust, where it was used to generate predictive ratings of movies, and TrustMail where the results are

used as a score for email messages. The data that allows these analyses to be performed and applications to be created all comes from the large repository of social network data on the Semantic Web.

While trust was the relationship feature analyzed here, the general technique of network analysis for developing algorithms is one we believe holds much promise. For example, a simplified version of the TrustMail application that utilizes basic social connectivity instead of trust ratings may be quite effective for filtering and scoring messages. Millions of people's social networks are represented in FOAF on the Semantic Web, and taking advantage of this large network as a source of application enhancing information holds promise for improving the usability and utility of many applications.

References

1. FOAF Vocabulary: http://xmlns.com/foaf/0.1/
2. Golbeck, Jennifer (2005) "Computing and Applying Trust in Web-Based Social Networks," Ph.D. Dissertation, University of Maryland, College Park.
3. Sinha, R., and Swearingen, K. (2001) "Comparing recommendations made by online systems and friends." *In Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries* Dublin, Ireland.
4. Swearingen, K. and R. Sinha. (2001) "Beyond algorithms: An HCI perspective on recommender systems," *Proceedings of the ACM SIGIR 2001 Workshop on Recommender Systems*, New Orleans, Louisiana.
5. Ziegler, Cai-Nicolas, Georg Lausen (2004) Analyzing Correlation Between Trust and User Similarity in Online Communities" *Proceedings of Second International Conference on Trust Management*, 2004.
6. Gil, Yolanda and Varun Ratnakar. (2002) "Trusting Information Sources One Citizen at a Time," *Proceedings of the First International Semantic Web Conference (ISWC)*, Sardinia, Italy.
7. Kamvar, Sepandar D. Mario T. Schlosser, Hector Garcia-Molina (2003) “The EigenTrust Algorithm for Reputation Management in P2P Networks”, *Proceedings of the 12th International World Wide Web Conference*, May 20-24, 2003, Budapest, Hungary.
8. Page, L., Brin, S., Motwani, R., & Winograd, T. (1998) “The PageRank citation ranking: Bringing order to the web.” *Technical Report 1998*, Stanford University, Stanford, CA.
9. Levin, Raph and Alexander Aiken. (1998)"Attack resistant trust metrics for public key certification." *7th USENIX Security Symposium*, San Antonio, Texas.
10. Richardson, Matthew, Rakesh Agrawal, Pedro Domingos. (2003) “Trust Management for the Semantic Web,” *Proceedings of the Second International Semantic Web Conference*. Sanibel Island, Florida.
11. Massa, P., P. Avesani. 2004. Trust-aware Collaborative Filtering for Recommender Systems. In *Proceedings of the International Conference on Cooperative Information Systems (CoopIS) 2004*.
12. Herlocker , Jonathan L., Joseph A. Konstan , Loren G. Terveen , John T. Riedl, (2004) Evaluating collaborative filtering recommender systems, *ACM Transactions on Information Systems (TOIS)*, v.22 n.1, p.5-53, January 2004.

Generalized Preferential Attachment: Towards Realistic Socio-Semantic Network Models

Camille Roth

CREA (Center for Research in Applied Epistemology),
CNRS/Ecole Polytechnique, 1 rue Descartes, 75005 Paris, France
roth@shs.polytechnique.fr

Abstract. The mechanism of preferential attachment underpins most recent social network formation models. Yet few authors attempt to check or quantify assumptions on this mechanism. We call *generalized preferential attachment* any kind of preference to interact with other agents with respect to any node property. We introduce tools for measuring empirically and characterizing comprehensively such phenomena, consequently suggest significant implications for model design, and apply these tools to a socio-semantic network of scientific collaborations, investigating in particular homophilic behavior. This opens the way to a whole class of realistic and credible social network morphogenesis models.

Keywords: Morphogenesis models, Preferential attachment, Social Networks, Dynamic Networks, Complex systems.

ACM: G.2.2; MSC: 68R10; PACS: 89.65.-s, 87.23.Ge, 89.75.-k

Introduction

A recent challenge in structural research in social science consists in modeling social network formation. Social networks are usually interaction networks — nodes are agents and links between nodes represent interactions between agents — and in this respect, modeling these networks involves disciplines linked both to graph theory (computer science and statistical physics), mathematical sociology and economics [1, 9, 28]. Most of the interest in this topic stems from the empirical observation that real social networks strongly differ from uniform random graphs as regards several statistical parameters; and foremost with respect to node connectivity distribution, or *degree* distribution. Indeed, in random graphs *a la* Erdos-Renyi [12] links between agents are present with a constant probability p and degree distributions follow a Poisson law, whereas empirical social networks exhibit power-law, or *scale-free*, degree distributions [1]. This phenomenon suggested that link formation does not occur randomly but rather

depends on node and network properties — that is, agents do not interact at random but instead according to heterogeneous preferences for other nodes.

Hence, early social network models endeavored to describe non-uniform interaction and growth mechanisms yielding the famous "scale-free" degree distribution [2]. Subsequently, much work has been focused on determining processes explaining and rebuilding more complex network structures consistent with those observed in the real world — a consistency validated through a rich set of statistical parameters measured on empirical networks, not limited to degree distribution but including as well clustering coefficient, average distance, assortativity, etc. [6, 7, 15, 22, 32].

However, even when cognitively, sociologically or anthropologically credible, most of the hypotheses driving these models are mathematical abstractions whose empirical measurement and justification are dubious, if any. In this paper, we call *preferential attachment* (PA) any kind of non-uniform interaction behavior and introduce tools for experimentally measuring PA with respect to any node property. We hence suggest significant implications for model design, and eventually apply these measures to an empirical case of socio-semantic network. In particular, we question degree-related PA, and estimate homophily.

1 A brief survey of social network models

Barabasi & Albert [2] pioneered the use of preferential linking in social network formation models to successfully rebuild a particular statistical parameter, the scale-free degree distribution. In their model, new nodes arrive at a constant rate and attach to already-existing nodes with a likeliness linearly proportional to their degree. This model has been widely spread and reused, and the term "preferential attachment" has consequently been often understood as degree-related only preferential attachment. Since then, many authors introduced diverse modes of preferential link creation depending on either various node properties (hidden variables and "types" [5, 30], fitness [7], centrality, euclidian distance [19, 13], common friends [17], bipartite structure [14], alleged underlying group structure [32], etc.) or on various linking mechanisms (competitive trade-off and optimization heuristics [10, 13], two-steps node choice [31], group formation [15, 24], to cite a few).

However and even in recent papers, hypotheses on PA are often arbitrary and at best supported by qualitative intuitions. Existing *quantitative* estimations of PA and consequent validations of modeling assumptions are quite rare, and either (i) related to the classical degree-related PA [3, 11, 16, 25], sometimes extended to a selected network property, like common acquaintances [21]; or (ii) reducing PA to a single parameter: for instance using direct mean estimation [15], econometric approaches [23] or Markovian models [29].[1] In addition, the way distinct properties correlatively influence PA is widely ignored. Thus, while of great interest in approaching the underlying behaviorial reality of social

[1] Let us also mention link prediction from similarity features based on various strictly structural properties [18], obviously somewhat related to PA.

networks, these works may not be able to provide a sufficient empirical basis and support for designing trustworthy PA mechanisms, and accordingly for proposing credible social network morphogenesis models. Yet in this view we argue that the following points are key:

1. Node degree does not make it all — and even the popular degree-related PA (a linear "rich-get-richer" heuristics) seems to be inaccurate for some types of real networks [3], and possibly based on flawed behavioral fundations, as we will suggest below in Sec. 4.2.
2. Strict social network topology and derived properties may not be sufficient to account for complex social phenomena — as several above-cited models suggest, introducing "external" properties (such as e.g. node types) may influence interaction; explaining for instance homophily-related PA [20] requires at least to *qualify* nodes using non-structural data.
3. Single parameters cannot express the rich heterogeneity of interaction behavior — for instance, when assigning a unique parameter to preferential interaction with close nodes, one misses the fact that such interaction could be significantly more frequent for very close nodes than for loosely close nodes, or discover that for instance it might be quadratic instead of linear with respect to the distance, etc.
4. Often models assume properties to be uncorrelated which, when it is not the case, would amount to count twice a similar effect;[2] knowing correlations between distinct properties is necessary to correctly determine their proper influence on PA.

To summarize, it is crucial to conceive PA in such a way that (i) it is a flexible and general mechanism, depending on relevant parameters based on both topological and non-topological properties; (ii) it is an empirically valid function describing the whole scope of possible interactions; and (iii) it takes into account overlapping influences of different properties.

2 Measuring preferential attachment

PA is the likeliness for a node to be involved in an interaction with another node with respect to node properties. In order to measure it, we first have to distinguish between (i) single node properties, or *monadic* properties (such as degree, age, etc.) and (ii) node dyad properties, or *dyadic* properties (social distance, dissimilarity, etc.). When dealing with monadic properties indeed, we seek to know the propension of some kinds of nodes to be involved in an interaction. On the contrary when dealing with dyads, we seek to know the propension for an interaction to occur preferentially with some kinds of couples.[3]

[2] Like for instance in [17] where effects related to degree and common acquaintances are combined in an independent way.

[3] Note that a couple of monadic properties can be considered dyadic; for instance, a couple of nodes of degrees k_1 and k_2 considered as a dyad (k_1, k_2). This makes the former case a refinement, not always possible, of the latter case.

2.1 Monadic PA

Suppose we want to measure the influence on PA of a given monadic property m taking values in $\mathcal{M} = \{m_1, ..., m_n\}$. We assume this influence can be described by a function f of m, independent of the distribution of agents of kind m. Denoting by "L" the event "attachment of a new link", $f(m)$ is simply the conditional probability $P(\mathrm{L}|m)$ that an agent of kind m is involved into an interaction.

Thus, it is $f(m)$ times more probable that an agent of kind m receives a link. We call f the *interaction propension* with respect to m. For instance, the classical degree-based PA used in Barabasi-Albert and subsequent models — links attach proportionally to node degrees [2, 3, 8] — is an assumption on f equivalent to $f(k) \propto k$.

$P(m)$ typically denotes the distribution of nodes of type m. The probability $P(m|\mathrm{L})$ for a new link extremity to be attached to an agent of kind m is therefore proportional to $f(m)P(m)$, or $P(\mathrm{L}|m)P(m)$. Applying the Bayes formula yields indeed:

$$P(m|\mathrm{L}) = \frac{f(m)P(m)}{P(\mathrm{L})} \tag{1}$$

with $P(\mathrm{L}) = \sum_{m' \in \mathcal{M}} f(m')P(m')$.

Empirically, during a given period of time ν new interactions occur and 2ν new link extremities appear. Note that a repeated interaction between two already-linked nodes is not considered a new link, for it incurs acquaintance bias. The expectancy of new link extremities attached to nodes of property m along a period is thus $\nu(m) = P(m|\mathrm{L}) \cdot 2\nu$. As $\frac{2\nu}{P(\mathrm{L})}$ is a constant of m we may estimate f through $\hat{f}$ such that:

$$\begin{cases} \hat{f}(m) = \dfrac{\nu(m)}{P(m)} & \text{if } P(m) > 0 \\ \hat{f}(m) = 0 & \text{if } P(m) = 0 \end{cases} \tag{2}$$

Thus $\mathbf{1}_P(m)f(m) \propto \hat{f}(m)$, where $\mathbf{1}_P(m) = 1$ when $P(m) > 0$, 0 otherwise.

2.2 Dyadic PA

Adopting a dyadic viewpoint is required whenever a property has no meaning for a single node, which is mostly the case for properties such as proximity, similarity — or distances in general. We therefore intend to measure interaction propension for a dyad of agents which fulfills a given property d taking values in $\mathcal{D} = \{d_1, d_2, ..., d_n\}$. Similarly, we assume the existence of an essential dyadic interaction behavior embedded into g, a strictly positive function of d; correspondingly the conditional probability $P(\mathrm{L}|d)$. Again, interaction of a dyad satisfying property d is $g(d)$ times more probable. In this respect, the probability for a link to appear between two such agents is:

$$P(d|\mathrm{L}) = \frac{g(d)P(d)}{P(\mathrm{L})} \tag{3}$$

with $P(\mathrm{L}) = \sum_{d' \in \mathcal{D}} g(d')P(d')$.

Here, the expectancy of new links between dyads of kind d is $\nu(d) = P(d|\mathrm{L})\nu$. Since $\frac{\nu}{P(\mathrm{L})}$ is a constant of d we may estimate g with $\hat{g}$:

$$\begin{cases} \hat{g}(d) = \dfrac{\nu(d)}{P(d)} & \text{if } P(d) > 0 \\ \hat{g}(d) = 0 & \text{if } P(d) = 0 \end{cases} \tag{4}$$

Likewise, we have $\mathbf{1}_P(d)g(d) \propto \hat{g}(d)$.

3 Interpreting interaction propensions

3.1 Shaping hypotheses

The PA behavior embedded in $\hat{f}$ (or $\hat{g}$) for a given monadic (or dyadic) property can be reintroduced as such in modeling assumptions, either (i) by reusing the exact empirically calculated function, or (ii) by stylizing the trend of $\hat{f}$ (or $\hat{g}$) and approximating f (or g) by more regular functions, thus making possible analytic solutions.

Still, an acute precision when carrying this step is often critical, for a slight modification in the hypotheses (e.g. non-linearity instead of linearity) makes some models unsolvable or strongly shakes up their conclusions. For this reason, when considering a property *for which there is an underlying natural order*, it may also be useful to examine the cumulative propension $\hat{F}(m_i) = \sum_{m'=m_1}^{m_i} \hat{f}(m')$ as an estimation of the integral of f, especially when the data are noisy (the same goes with $\hat{G}$ and $\hat{g}$).

3.2 Correlations between properties

Besides, if modelers want to consider PA with respect to a collection of properties, they have to make sure that the properties are uncorrelated or that they take into account the correlation between properties: evidence suggests indeed that for instance node degrees depend on age. If two distinct properties p and p' are independent, the distribution of nodes of kind p in the subset of nodes of kind p' does not depend on p', i.e. the quantity $\frac{P(p|p')}{P(p)}$ must theoretically be equal to 1, $\forall p, \forall p'$. Empirically, it is possible to estimate it through:[4]

$$\begin{cases} \widehat{c_{p'}}(p) = \dfrac{P(p|p')}{P(p)} & \text{if } P(p) > 0 \\ \widehat{c_{p'}}(p) = 0 & \text{if } P(p) = 0 \end{cases} \tag{5}$$

in the same manner as previously.

[4] For computing the correlation between a monadic and a dyadic property, it is easy to interpret $P(p|d)$ as the distribution of p-nodes being part of a dyad d.

3.3 Essential behavior

As such, calculated propensions do not depend on the distribution of nodes of a given type at a given time. In other words, if for example physicists prefer to interact twice more with physicists than with sociologists but there are three times more sociologists around, physicists may well be apparently interacting more with sociologists. Nevertheless, $\hat{f}$ remains free of such biases and yields the "baseline" preferential interaction behavior of physicists.

However, $\hat{f}$ could still depend on global network properties, e.g. its size, or its average shortest path length. Validating the assumption that $\hat{f}$ is independent of *any* global property of the network — i.e., that it is an *entirely essential* property of nodes of kind p — would require to compare different values of $\hat{f}$ for various periods and network configurations. Put differently, this entails checking whether the shape of $\hat{f}$ itself is a function of global network parameters.

3.4 Activity

Additionally, $\hat{f}$ represents equivalently an attractivity or an activity: if interactions occur preferentially with some kinds of agents, it could as well mean that these agents are more attractive or that they are more active. If more attractive, the agent will be interacting more, thus being apparently more active. To distinguish between the two effects, it is sometimes possible to measure independently agent activity, notably when interactions occur during *events*, or when interaction initiatives are traceable (e.g. in a directed network).

In such cases, the distinction is far from neutral for modeling. Indeed, when considering evolution mechanisms focused not on agents creating links, but instead on events gathering agents (like in [15, 24]), modelers have to be careful when integrating back into models the observed PA as a behavioral hypothesis. Some categories of agents might in fact be more active and accordingly involved in more events, not enjoying more attractivity. This would eventually lead the modeler to refine agent behavior characterization by including both the participation in events and the number of interactions per event, rather than just preferential interactions.

4 An application to socio-semantic networks

4.1 Definitions

We now apply the above tools to a socio-semantic network, that is, a social network where agents are also linked to semantic items. We examine therein two particular kinds of PA: (i) PA related to a monadic property: the node degree; and (ii) PA linked to a dyadic property: homophily, i.e. the propension of individuals to interact more with similar agents.

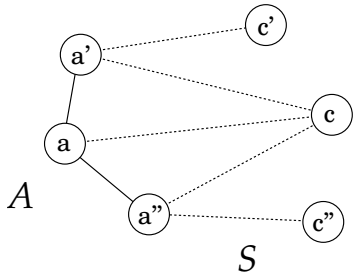

Fig. 1. Sample socio-semantic network (3 agents a, a', a'' and 3 concepts c, c', c'').

Networks. The social network $\mathcal{A}$ is the network of agents, where links correspond to interactions: $\mathcal{A} = (\mathbf{A}, E_{\mathbf{A}})$, with $\mathbf{A}$ denoting the agent set and $E_{\mathbf{A}}$ the (undirected) set of links between agents. Interactions occur through events, and each event is associated with a semantic content, made of semantic markers (e.g. keywords), or *concepts* taken in a concept set $\mathbf{C}$. Similarly, agents are linked to concepts associated with events they are involved in, forming a second network, $\mathcal{S} = (\mathbf{A} \cup \mathbf{C}, E_{\mathbf{AC}})$. Thus we deal with two kinds of links: (i) between pairs of agents, and (ii) between concepts and agents. Since we measure agent behavior through network dynamics, we also consider the temporal series of networks $\mathcal{A}(t)$ and $\mathcal{S}(t)$, with $t \in \mathbb{N}$, which altogether make a dynamic socio-semantic network (see Fig. 1).

Empirical protocol. Empirical data come from the bibliographical database Medline which contains dated abstracts of published articles of biology and/or medicine. We focused on a portion concerning a well-defined community of embryologists working on the *zebrafish*, during the period 1997-2004. Translated in the above framework, articles are events, their authors are the agents, and semantic markers are made of expert-selected abstract words. In order to have a non-empty and statistically significant network for computing propensions, we first build the network on an initialization period of 7 years (from 1997 to end-2003), then carry the calculation on new links appearing during the last year. The dataset contains around $10,000$ authors, $5,000$ articles and 70 concepts; about $10,000$ new links appear during the last year.

4.2 Degree-related PA

We use Eq. 2 and consider the node degree k as property m (thus $\mathcal{M} = \mathbb{N}$): in this manner, we intend to compute the real slope $\hat{f}(k)$ of the degree-related PA and compare it with the assumption "$f(k) \propto k$". This hypothesis classically relates to the preferential linking of new nodes to old nodes. To ease the comparison, we considered the subset of interactions between a new and an old node.

Empirical results are shown on Fig. 2. Seemingly, the best linear fit corroborates the data and tends to confirm that $f(k) \propto k$. The best non-linear fit

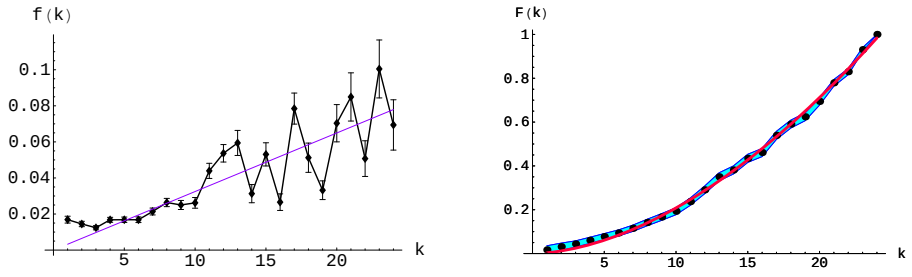

Fig. 2. *Left:* Degree-related interaction propension $\hat{f}$, computed on a one-year period, for $k < 25$ (confidence intervals are given for $p < .05$); the solid line represents the best linear fit. *Right:* Cumulated propension $\hat{F}$. Dots represent empirical values, the solid color line is the best non-linear fit for $\hat{F} \sim k^{1.83}$, and the gray area is the confidence interval.

however deviates from this hypothesis, suggesting that $f(k) \propto k^{0.97}$. However, the confidence interval on this exponent is $[0.6 - 1.34]$ thus dramatically too wide to determine the precise exponent, which may be critical. When the data is noisy like in the present situation, since there is a natural order on k it is very instructive to plot the cumulated propension $\hat{F(k)} = \sum_{k'=1}^{k} \hat{f}(k)$ on Fig. 2. In this case, the best non-linear fit for $\hat{F}$ is $\hat{F}(k) \propto k^{1.83}$ ± 0.05, confirming the slight deviation from a strictly linear preference which would yield k^2.

Rich-work-harder. This precise result is not new and agrees with existing studies of the degree-related PA (e.g. [16, 21]). Nevertheless, we wish to stress a more fundamental point concerning this kind of PA. Indeed, considerations on agent activity lead us to question the usual underpinnings and justifications of PA related to a monadic property. Regarding in particular degree-related PA, we question the *"rich-get-richer"* metaphor describing rich, or well-connected agents as more attractive than poorly connected agents, thus receiving more connections and becoming even more connected.[5]

When considering the activity of agents with respect to k, that is, the number of events in which they participate (here, the number of articles they co-author), "rich" agents are proportionally more active than "poor" agents (see Fig. 3), and thus obviously encounter more interactions. It might thus well simply be that richer agents work harder, not are more attractive; the underlying behavior linked to preferential interaction being simply "proportional activity".[6]

[5] "(...) the probability that a new actor will be cast with an established one is much higher than that the new actor will be cast with other less-known actors" [2].

[6] We obviously make the assumption that k accurately reflects author activity, i.e. a behavioral feature. Thus k is a proxy for agent activity and, if the number of coauthors does not depend on k (which is actually roughly the case in this data), then observing a quasi-linear degree-related PA is not surprising.

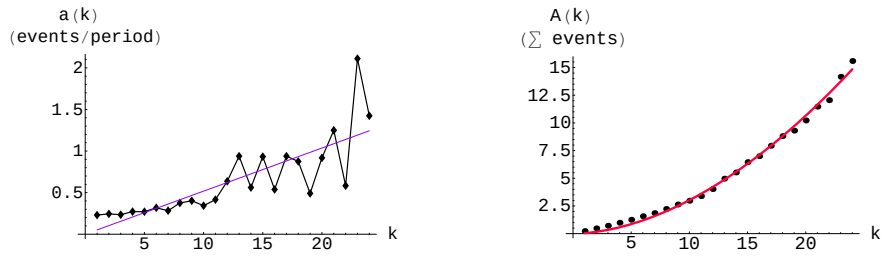

Fig. 3. *Left:* Activity $a(k)$ during the same period, in terms of articles per period (events per period) with respect to agent degree; solid line: best linear fit. *Right:* Cumulated activity $A(k) = \sum_{k'=1}^{k} a(k)$, best non-linear fit is $k^{1.88}$ ± 0.09.

While formally equivalent from the viewpoint of PA measurement, the *"rich-get-richer"* and *"rich-work-harder"* metaphors are not behaviorally equivalent. One could choose to be blind to this phenomenon and keep an interaction propension proportional to node degree. On the other hand, one could also prefer to consider higher-degree nodes as more active, assuming instead that the number of links per event is degree-independent and that agents do neither *prefer*, nor *decide* to interact with famous, highly connected nodes; a hypothesis supported by the present empirical results. These two viewpoints, while both consistent with the observed PA, bear distinct implications for modeling as underlined in Sec. 3.4.

More generally, such feature supports the idea that events, not links, are the right level of modeling for social networks — with events reducing in some cases to a dyadic interaction. Then, modelers would have to break down interaction propensions into (i) activities (number of events) and (ii) interactivities (number of interactions per event).

4.3 Homophilic PA

Homophily translates the fact that agents prefer to interact with other resembling agents. Here, we assess the extent to which agents are "homophilic" by introducing an inter-agent semantic distance. By semantic distance we mean a function of a dyad of nodes that enjoys the following properties: (i) decreasing with the number of shared concepts between the two nodes, (ii) increasing with the number of distinct concepts, (iii) equal to 1 when agents have no concept in common, and to 0 when they are linked to identical concepts.

Given $(a, a') \in \mathbf{A}^2$ and denoting by $a^\wedge$ the set of concepts a is linked to, we introduce a semantic distance $\delta(a, a') \in [0; 1]$ satistying the previous properties:[7]

$$\delta(a, a') = \frac{|(a^\wedge \setminus a'^\wedge) \cup (a'^\wedge \setminus a^\wedge)|}{|a^\wedge \cup a'^\wedge|}$$

As δ takes real values in $[0, 1]$ we need to discretize δ. To this end, we use a uniform partition of $[0; 1[$ in I intervals, to which we add the singleton $\{1\}$. We thus define a new discrete property d taking values in $\mathcal{D} = \{d_0, d_1, ..., d_I\}$ consisting of $I+1$ intervals: $\mathcal{D} = \left\{[0; \frac{1}{I}[;\ [\frac{1}{I}; \frac{2}{I}[; ...[\frac{I-1}{I}; 1[; \{1\}\right\}$. Finally, we obtain an empirical estimation of homophily with respect to this distance by applying Eq. 4 on d, with $I = 15$.

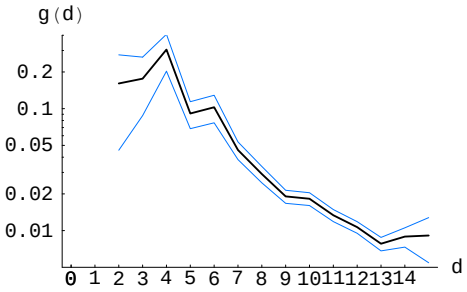

Fig. 4. *Thick solid line:* Homophilic interaction propension $\hat{g}$ with respect to $d \in \mathcal{D} = \{d_0, ..., d_{15}\}$. *Thin lines:* Confidence interval, for $p < .05$. Because several fitting functions are conceivable no particular fitting has been carried on this graph. The y-axis is in log-scale.

The results are gathered on Fig. 4 and show that while agents favor interactions with slightly different agents (as the initial increase suggests), they still very strongly prefer similar agents, as the clearly decreasing trend indicates (sharp decrease from d_4 to d_{13}, with d_4 being one order of magnitude larger than d_{13} — note also that $\hat{g}(d_0) = \hat{g}(d_1) = 0$ because no new link appears for these distance values). Agents thus display semantic homophily, a fact that fiercely advocates

[7] Note that this kind of distance, based on the Jaccard coefficient [4], has been extensively used in Information Retrieval, as well as recently for link formation prediction in [18]. The point here is however not to focus on this particular similarity measure, but to show that simple non-topological properties may also strongly influence interaction behavior.

Written in a more explicit manner, with $a^\wedge = \{c_1, ..., c_n, c_{n+1}, ..., c_{n+p}\}$ and $a'^\wedge = \{c_1, ..., c_n, c'_{n+1}, ..., c'_{n+q}\}$, we have $\delta(a, a') = \frac{p+q}{p+q+n}$; n and p, q representing respectively the number of elements $a^\wedge$ and $a'^\wedge$ have in common and have in proper. We also verify that if $n = 0$ (disjoint sets), $\delta(a, a') = 1$; if $n \neq 0$, $p = q = 0$ (same sets), $\delta(a, a) = 0$; and if $a^\wedge \subset a'^\wedge$ (included sets), $\delta(a, a') = \frac{q}{q+n}$. It is moreover easy though cumbersome to show that $\delta(.,.)$ is also a *metric* distance.

the necessity of taking semantic content into account when modeling such social networks.

4.4 Correlation between degree and semantic distance

In other words, the exponential trend of $\hat{g}$ suggests that scientists seem to choose collaborators most importantly because they are sharing interests, and less because they are attracted to well-connected colleagues, which besides actually seems to reflect agent activity. As underlined in Sec. 3.1, when building a model of such network based on degree-related and homophilic PA, one has to check whether the two properties are independent, i.e. whether or not a node of low degree is more or less likely to be at a large semantic distance of other nodes. It appears here that there is no correlation between degree and semantic distance: for a given semantic distance d, the probability of finding a couple of nodes including a node of degree k is the same as it is for any value of d — see Fig. 5.

To go further, we might suggest that socio-semantic networks are structured in communities because agents group according to similar interests, in epistemic communities [27], through a mechanism involving events where agents are more or less active, and gather preferentially with respect to their interests; the former being entirely independent of the latter.

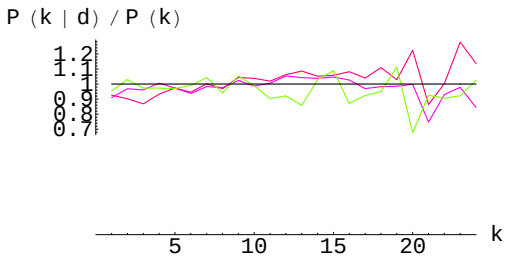

Fig. 5. Degree and semantic distance correlation estimated through $\widehat{c_d}(k) = \dfrac{P(k|d)}{P(k)}$, plotted here for three different values of d: $d \in \{d_5, d_8, d_{11}\}$.

Conclusion

Preferential attachment is the cornerstone of growth mechanisms in most recent social network formation models. This notion was established by the success of a pioneer model [2] rebuilding a major stylized fact of empirical networks, the scale-free degree distribution. While PA has subsequently been widely used, few authors have tried to check or quantify the rather arbitrary assumptions on PA — when such prospects exist, they are mostly dealing with degree-related

PA or estimating PA phenomena as single parameters. Models should go further towards empirical investigation when designing hypotheses. This would be really appealing to social scientists, who are usually not seeking *normative* models. We are confident the present reluctance to measuring interaction behaviors and processes is due to the lack of a clean general framework for this purpose, which is the aim of this paper.

We thus introduced the notion of "interaction propension", whereby we assume that agents have an *essential* preferential interaction behavior. Using this concept, we designed measurement tools for quantifying, in a dynamic network, any kind of PA with respect to any property of a single node or of a dyad of nodes — a generalized preferential attachment. The result is a function yielding a comprehensive description of interaction behavior related to a given property. In addition to clarifying PA three features are crucial: (i) properties not related to the network structure (such as homophily), (ii) correlations between properties, (iii) activity of agents and nature of interactions (i.e. modeling events, not nodes attaching to each other). This kind of hindsight on the notion and status of PA should be useful, even for normative models.

We finally applied these tools to a particular case of socio-semantic network, a scientific collaboration network with agents linked to semantic items. While we restricted ourselves to a reduced example of two significant properties (node degree and semantic distance), measuring PA relatively to other parameters could actually have been very relevant as well — such as PA based on social distance for instance (shortest path length between two agents in the social network). Specifying the list of properties is nevertheless a process driven by the real-world situation *and* by the stylized facts the modeler aims at rebuilding and considers relevant for morphogenesis.

More generally, this framework could be applied to any kind of network (including semantic web formation modeling) as well as adapted to disconnection propensions. Likewise, once propensions of *interaction in the broad sense* are known, a whole class of morphogenesis models [5, 9, 26] can be designed, with agents interacting on a growing network according to stylized interaction heuristics, heuristics precisely based on those measured empirically. *In fine*, introducing more credible hypotheses based on real-case empirical measures would obviously help attract more social scientists in this promising field.

Acknowledgements. The author wishes to thank Clémence Magnien, Matthieu Latapy and Paul Bourgine for very fruitful discussions, and also acknowledges interesting remarks from David Chavalarias and three anonymous reviewers. This work has been partially funded by the CNRS.

References

1. R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, 2002.
2. A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

3. A.-L. Barabási, H. Jeong, R. Ravasz, Z. Neda, T. Vicsek, and T. Schubert. Evolution of the social network of scientific collaborations. *Physica A*, 311:590–614, 2002.
4. V. Batagelj and M. Bren. Comparing resemblance measures. *Journal of Classification*, 12(1):73–90, 1995.
5. M. Boguna and R. Pastor-Satorras. Class of correlated random networks with hidden variables. *Physical Review E*, 68:036112, 2003.
6. M. Boguna, R. Pastor-Satorras, A. Diaz-Guilera, and A. Arenas. Models of social networks based on social distance attachment. *Physical Review E*, 70:056122, 2004.
7. G. Caldarelli, A. Capocci, P. D. L. Rios, and M. A. Munoz. Scale-free networks from varying vertex intrinsic fitness. *Physical Review Letters*, 89(25):258702, 2002.
8. M. Catanzaro, G. Caldarelli, and L. Pietronero. Assortative model for social networks. *Physical Review E*, 70:037101, 2004.
9. P. Cohendet, A. Kirman, and J.-B. Zimmermann. Emergence, formation et dynamique des réseaux – modèles de la morphogenèse. *Revue d'Economie Industrielle*, 103(2-3):15–42, 2003.
10. V. Colizza, J. R. Banavar, A. Maritan, and A. Rinaldo. Network structures from selection principles. *Physical Review Letters*, 92(19):198701, 2004.
11. E. Eisenberg and E. Y. Levanon. Preferential attachment in the protein network evolution. *Physical Review Letters*, 91(13):138701, 2003.
12. P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.
13. A. Fabrikant, E. Koutsoupias, and C. H. Papadimitriou. Heuristically optimized trade-offs: A new paradigm for power laws in the internet. In *ICALP '02: Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, pages 110–122, London, UK, 2002. Springer-Verlag.
14. J.-L. Guillaume and M. Latapy. Bipartite structure of all complex networks. *Information Processing Letters*, 90(5):215–221, 2004.
15. R. Guimera, B. Uzzi, J. Spiro, and L. A. N. Amaral. Team assembly mechanisms determine collaboration network structure and team performance. *Science*, 308:697–702, 2005.
16. H. Jeong, Z. Néda, and A.-L. Barabási. Measuring preferential attachment for evolving networks. *Europhysics Letters*, 61(4):567–572, 2003.
17. E. M. Jin, M. Girvan, and M. E. J. Newman. The structure of growing social networks. *Physical Review E*, 64(4):046132, 2001.
18. D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 556–559, New York, NY, USA, 2003. ACM Press.
19. S. S. Manna and P. Sen. Modulated scale-free network in euclidean space. *Physical Review E*, 66:066114, 2002.
20. M. McPherson and L. Smith-Lovin. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–440, 2001.
21. M. E. J. Newman. Clustering and preferential attachment in growing networks. *Physical Review Letters E*, 64(025102), 2001.
22. M. E. J. Newman. The structure of scientific collaboration networks. *PNAS*, 98(2):404–409, 2001.
23. W. W. Powell, D. R. White, K. W. Koput, and J. Owen-Smith. Network dynamics and field evolution: The growth of interorganizational collaboration in the life sciences. *American Journal of Sociology*, 110(4):1132–1205, 2005.

24. J. J. Ramasco, S. N. Dorogovtsev, and R. Pastor-Satorras. Self-organization of collaboration networks. *Physical Review E*, 70:036106, 2004.
25. S. Redner. Citation statistics from 110 years of physical review. *Physics Today*, 58:49–54, 2005.
26. C. Roth and P. Bourgine. Binding social and cultural networks: a model. arXiv.org e-print archive, nlin.AO/0309035, 2003.
27. C. Roth and P. Bourgine. Epistemic communities: Description and hierarchic categorization. *Mathematical Population Studies*, 12(2):107–130, 2005.
28. B. Skyrms and R. Pemantle. A dynamic model of social network formation. *PNAS*, 97(16):9340–9346, 2000.
29. T. A. Snijders. The statistical evaluation of social networks dynamics. *Sociological Methodology*, 31:361–395, 2001.
30. B. Söderberg. A general formalism for inhomogeneous random graphs. *Physical Review E*, 68:026107, 2003.
31. H. Stefancic and V. Zlatic. Preferential attachment with information filtering–node degree probability distribution properties. *Physica A*, 350(2-4):657–670, 2005.
32. D. J. Watts, P. S. Dodds, and M. E. J. Newman. Identity and search in social networks. *Science*, 296:1302–1305, 2002.

Network Analysis as a Basis for Partitioning Class Hierarchies

Heiner Stuckenschmidt

Vrije Universiteit Amsterdam
de Boelelaan 1081a, 1081HV Amsterdam
heiner@cs.vu.nl

Abstract. We discuss the use of network analysis methods to support the automatic partitioning of large concept hierarchies. Different from other work in the area, we directly apply these methods on the structure of the hierarchy. We show that this way of using network analysis techniques can provide significant results with respect to identifying key concepts and using them to determine subsets of class hierarchies that are related content-wise. We discuss the methods used and evaluate the result on the ACM classification of computer science topis.

1 Motivation

Network analysis techniques are recently receiving some attention in the area of semantic web research. People have recognized that information about communication patterns and social relationships can be used to better understand and design system behavior and to provide more efficient information sharing. Most existing work applies network analysis techniques in a rather standard way by analyzing relations between people (often users or their representation). Some work tries to enrich social networks with an analysis of shared topics or shared terminology [5]. The work reported in this paper takes a different approach. Rather than analyzing people, we use network analysis methods to analyze ontologies. For this purpose, we treat ontologies as networks where named elements in the ontology (concepts, relations, instances) are nodes. Links between these named elements are derived from the definitions and axioms in the ontology. Similar ideas are reported in [4].

Based on this network representation of an ontology, available analysis techniques provide us with a wide range of possibilities such as

- determining important concepts
- valuating the degree of dependency between two concepts
- finding sets of related concepts
- determining completely unrelated parts of an ontology
- etc.

In this paper, we focus on the use of network analysis for a particular task: the partitioning of an ontology into a number of disjoint and covering set of concepts. There are a number of use cases for this kind partitioning including distributed maintenance,

selective reuse [6] and efficient reasoning [1]. We have developed a method for partitioning large taxonomies based on the structure of the hierarchy [7]. We describe this method and extend the work reported in [7] in two ways:

- We propose a heuristics for improving the result of the partitioning method by merging strongly related subparts
- We present results from an empirical evaluation of the methods on the ACM classification of computer science topics

The paper is structured as follows. In section 2 we provide a brief overview of the partitioning method in terms of several steps that lead from a given concept hierarchy to an assignment of concepts to modules. In section 3 we discuss the network analysis methods used in our approach in more detail and give examples for their application. Section 4 reports the setting and results from an evaluation of the partitioning method. We conclude with a discussion of the method and the general idea of using network analysis methods for analyzing ontologies.

2 Overview of the Partitioning Method

In [7] we presented a method for automatically partitioning lightweight ontologies. In particular, the method was aimed at models that only consist of a concept hierarchy. We showed that using simple heuristics, we can create meaningful partitions of class hierarchies for the purpose of supporting browsing and visualization of large hierarchies. We briefly recapitulate the different steps of our method as the following discussions will be based on this information.

Step 1: Create Dependency Graph: In the first step a dependency graph is extracted from an ontology source file. The idea is that elements of the ontology (concepts, relations, instances) are represented by nodes in the graph. Links are introduced between nodes if the corresponding elements are related in the ontology, e.g. because they appear in the same definition.

Step 2: Determine strength of Dependencies: In the second step the strength of the dependencies between the concepts has to be determined. This actually consists of two parts: First of all, we can use algorithms from network analysis to compute degrees of relatedness between concepts based on the structure of the graph. Second, we can use weights to determine the importance of different types of dependencies, e.g. we can decide subclass relations have a higher impact than domain relations[1].

Step 3: Determine Modules The proportional strength network provides us with a foundation for detecting sets of strongly related concepts. This is done using a graph algorithm that detects minimal cuts in the network and uses them to split the

[1] In the following we are dealing with the subclass relation as the only type on dependency and therefore ignore the concept of weights

overall graph in sets of nodes that are less strongly connected to nodes outside the set than to nodes inside.

Step 4/5: Improving the Partitioning In the last steps the created partitioning is optimized. In these steps nodes leftover nodes from the previous steps are assigned to the module they have the strongest connection to. Further, we merge smaller modules into larger ones to get a less scattered partitioning. Candidates for this merging process are determined using a measure of coherence.

3 Using Network Analysis Methods

The main rational for translating ontologies into a graph structure is the availability of sophisticated methods for analyzing graph structures. Experiments with different available methods for determining the strength of relationships between nodes in a graph as well as different methods for partitioning graphs into disjoint sets of nodes revealed that the use of the line island method [2] on a relative strength network provides useful results. In the following, we present the network analysis methods used in our approach, results of applying them on an existing ontology about public and private transportation are shown in the next section.

Essentially, network analysis methods are used for determining the strength of relationships between nodes in the graph (step 2), for identifying potential partitions (step 3) and for improving the partitioning by determining parts to be merged (step 5).

3.1 Determining Strength of Dependencies: Relative Strength

After the dependency graph for an ontology has been created in the way descried in the last section the strengths of the dependencies between the concepts have to be determined. Following the basic assumption of our approach, we use the structure of the dependency graph to determine the weights of dependencies. In particular we use results from social network theory by computing the proportional strength network for the dependency graph. The strength of the dependency of a connection between a node c_i and c_j is determined to be the proportional strengths of the connection. The proportional strength describes the importance of a link from one node to the other based on the number of connections a node has. In general it is computed by dividing the sum of the weights of all connections between c_i and c_j by the sum of the weights of all connections c_i has to other nodes (compare [3], page 54ff):

$$w(c_i, c_j) = \frac{w_{ij} + w_{ji}}{\sum_k w_{ik} + w_{ki}}$$

Here w_{ij} is the weight preassigned to the link between c_i and c_j - in the experiments reported below this will always be one. As a consequence, the proportional strength used in the experiments is one divided by the number of nodes c_i is connected to. The intuition behind it is that individual social contacts become more important if there are

only few of them. In our setting, this measure is useful because we want to prevent that classes that are only related to a low number of other classes get separated from them. This would be against the intuition that classes in a module should be related. We use node **d** in Figure 1 to illustrate the calculation of weights using the proportional

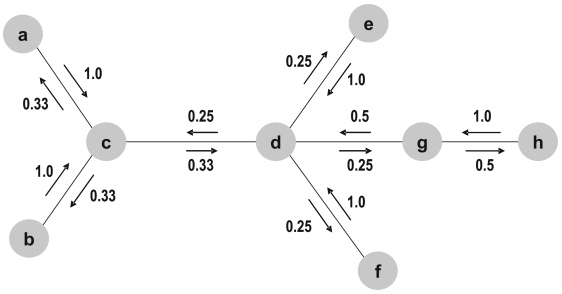

Fig. 1. An Example Graph with proportional strength dependencies

strength. The node has four direct neighbors, this means that the proportional strength of the relation to these neighbors is 0.25 (one divided by four). Different levels of dependency between **d** and its neighbors now arise from the relative dependencies of the neighbors with **d** (the proportional strength is non-symmetric). We see that **e** and **f** having no other neighbors completely depend on **d**. The corresponding value of the dependency is 1. Further, the strength of the dependency between **g** and **d** is 0.5, because **g** has two neighbors and the dependency between **b** and **d** is 0.33 as **b** has 3 neighbors.

3.2 Determine Modules: Line Islands

The proportional strength network provides us with a foundation for detecting sets of strongly related concepts that are candidates for forming a part in the partition. For this purpose, we make use of an algorithm that computes all maximal line islands of a given size in a graph [2].

Definition 1 (Line Island). *A set of vertices $I \subseteq C$ is a line island in a dependency graph $G = (C, D, w)$ - where C is a set of nodes, D a set of edged representing dependencies between them and w is a set of weights of the edges - if and only if*

- *I induces a connected subgraph of G*
- *There is a weighted graph $T = (V_T, E_T, w_T)$ such that:*
  - *T is embedded in G*
  - *T is an maximal spanning tree with respect to I*
  - *the following equation holds:*

$$\max_{\{(v_i,v_j)\in D|(v_i\in I\wedge v_j\notin I)\vee(v_j\in I\wedge v_i\notin I)\}} w_{ij} < \min_{(v_k,v_l)\in E_T} w_{kl})$$

Note that for the determination of the maximal spanning tree the direction of edges is not considered.

This criterion exactly coincides with our intuition about the nature of modules given in the introduction, because it determines sets of concepts that are stronger internally connected than to any other concept not in the set. The algorithm requires an upper and a lower bound on the size of the detected set as input and assigns an island number to each node in the dependency graph. We denote the island number assigned to a concept c as $\alpha(c)$. The assignment $\alpha(c) = 0$ means that c could not be assigned to an island.

We use different sets of nodes in the graph in Figure 1 to illustrate the concept of a line island. Let us first consider the set $\{\mathbf{a}, ..., \mathbf{f}\}$. It forms a connected subgraph. The maximal spanning tree of this set consists of the edges $\mathbf{a} \xrightarrow{1.0} \mathbf{c}$, $\mathbf{b} \xrightarrow{1.0} \mathbf{c}$, $\mathbf{c} \xrightarrow{0.33} \mathbf{d}$, $\mathbf{e} \xrightarrow{1.0} \mathbf{d}$, and $\mathbf{f} \xrightarrow{1.0} \mathbf{d}$. We can see however, that this node set is not an island, because the minimal weight of an edge in the spanning tree is 0.33 and there is an incoming edge with strength 0.5 ($\mathbf{g} \xrightarrow{0.5} \mathbf{d}$). If we look at the remaining set of nodes $\{\mathbf{g}, \mathbf{h}\}$, we see that it fulfills the conditions of an island: it forms a connected subgraph, the maximal spanning tree consists of the edge $\mathbf{h} \xrightarrow{1.0} \mathbf{g}$ and the maximal value of in- or outgoing links is 0.5 ($\mathbf{g} \xrightarrow{0.5} \mathbf{d}$). This set, however, is not what we are looking for because it is not maximal: it is included in the set $\{\mathbf{d}, ..., \mathbf{h}\}$. This set is a line island with the maximal spanning tree consisting of the edges $\mathbf{e} \xrightarrow{1.0} \mathbf{d}$, $\mathbf{f} \xrightarrow{1.0} \mathbf{d}$, $\mathbf{g} \xrightarrow{0.5} \mathbf{d}$ and $\mathbf{h} \xrightarrow{1.0} \mathbf{g}$ where the minimal weight (0.5) is higher than the maximal weight of any external link which is $\mathbf{c} \xrightarrow{0.33} \mathbf{d}$. Another reason for preferring this island is that the remaining node set $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ also forms a line island with maximal spanning tree $\mathbf{a} \xrightarrow{1.0} \mathbf{c}$, $\mathbf{b} \xrightarrow{1.0} \mathbf{c}$ and the weaker external link $\mathbf{c} \xrightarrow{0.33} \mathbf{d}$.

3.3 Improving Partitions: Height of Line Islands

A problem of the use of the line island method for determining partitions is that it often creates a number of very small parts that only consists of two or three concepts. When inspecting the dependencies in the relevant parts of the hierarchy, we discovered that most of the problematic modules have very strong internal dependencies. In order to distinguish such cases, we need a measure for the strength of the internal dependency. The measure that we use is called the 'height' of an island. It uses the minimal spanning tree T used to identify the module: the overall strength of the internal dependency equals the strength of the weakest link in the spanning tree.

$$height(P) = \min_{(v_i,v_j)\in E_T} w_{ij}$$

We can again illustrate the the concept of height using the example from figure 1. We identifies two islands, namely $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ and $\{\mathbf{d}, ..., \mathbf{h}\}$. As the maximal spanning tree of the first island consists of the two edges $\mathbf{a} \xrightarrow{1.0} \mathbf{c}$, $\mathbf{b} \xrightarrow{1.0} \mathbf{c}$, the height of this

island is 1.0. In the maximal spanning tree of the second island the edge $\mathbf{g} \overset{0.5}{\rightarrow} \mathbf{d}$ is the weakest link that therefore sets the height of the island to 0.5.

The height of the island provides us with a criterion for automatically selecting parts of the graph to be merged again. In a series of experiments, we observed that most islands that with a height of 0.5 or more do not not correspond to meaningful modules and are therefore candidates for merging with other islands. A second choice that has to be made is about which other island to merge with. There are two factors that influence this decision. The first is the height of the other module. Here we prefer to merge with other islands that have a high height as well. The second factor is the strength of the relation between the two islands. We determine this strength by adding all edges that exist between nodes in the two islands. Based on these two factors, we determine the merging potential $m_{P_1}(P_2)$ of islands P_1 with island P_2 as follows:

$$m_{P_1}(P_2) = height(P_2) \cdot \sum_{v_i \in P_1, v_j \in P_2} w_{ij} + w_{ji}$$

Candidates for merging are now determined by ordering al islands based on their height. For each island with a height of 0.5 or more, we compute the merging potential with respect to all other islands. The island is merged with the one that has the highest merging potential. If the potential for different islands is the same, we chose the one with the highest height.

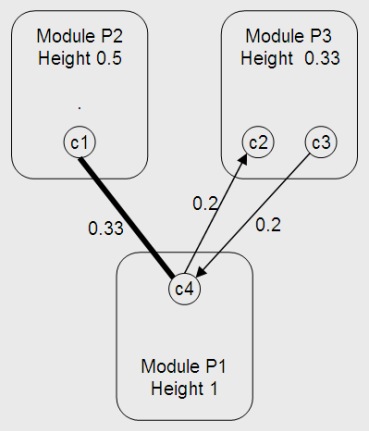

Fig. 2. Example of a merging decision

Figure 2 shows a simple example consisting of three islands (modules). Ordering the modules based on their height we get the sequence P_1, P_2, P_3. We start with P_1

which has the highest height and compute the merging potential with respect to P_2 and P_3 as follows:

$$m_{P_1}(P_2) = 0.5 \cdot 0.33 = 0.165$$
$$m_{P_1}(P_3) = 0.33 \cdot (0.2 + 0.2) = 0.132$$

As the merging potential with respect to P_2 is higher that the one with respect to P_3, we merge the two islands. As a result, the height of this new set of nodes becomes 0.33, because the edge between the two islands is now the one with the lowest weight. This means that we end up with two modules of height 0.33. No more merging operations are required.

4 Empirical Evaluation

We consider an imaginary optimal partitioning of the ontology. An automatically generated partitioning is evaluated against this optimal partitioning. The basis for comparison are pairs of classes. In particular, we consider pairs of concepts that are in the same part of the model These pairs are called intra-pairs, respectively. Based on the notion of intra-pairs, we can define two quality measures for a generated partitioning.

Precision The precision of a partitioning is defined as the ratio of intra-pairs in the generated partitioning that are also intra-pairs in the optimal partitioning.

Recall The recall of a partitioning is defined by the ratio of intra-pairs in the optimal partitioning that are also in the generated one.

The basic problem of evaluating a partitioning is the fact, that in most cases we do not have an optimal partitioning to compare to. For these cases, we have to rely on alternative methods to determine the quality of the partitioning. A possibility that we will explore is empirical evaluation through user testing. Such an evaluation requires that the subjects have some knowledge about the domain modeled by the ontology. Therefore the ontology and the subjects have to be chosen carefully. The first option is to chose an ontology about a rather general topic (e.g. the transportation ontology). In this case any student is knowledgable enough to be chosen as a test subject. The other option is to chose a more specialized model and look for domain experts. Options here are the use of a computer science specific ontology (eg. the ACM classification) or a medical ontology. The advantage of the former is that test subjects are easier available while the time of medical experts if often rather limited.

A basic problem of empirical evaluation is the complexity of the task. Users will often not be able to oversee the complete ontology and to determine a good partitioning for themselves (in fact this is the reason why we need automatic partitioning).The most basic way of doing empirical evaluation is to directly use the notion of intra-pairs. As we have seen above, knowing all intra-pairs is sufficient for determining the quality measures defined above. This means that we can present pairs of concepts to subjects and ask them whether or not these concepts should be in the same part of the ontology. A problem of this approach is that the subject is not forced to be consistent. It might

happen, that according to a subject A and B as well as A and C should be in the same part, but B and C should not. The second problem is the number of tests necessary to determine a partitioning. In the case of the ACM hierarchy, more that 1,5 Million pairs would have to be tested. In order to avoid these problems of consistency and scalability of empirical evaluation, we decided to perform an evaluation that is not based on concept pairs. The setting of our experiments is described in the following.

4.1 Setting

We used the ACM classification of computer science topics as a basis for performing an empirical evaluation of our partitioning method. The nature of the ACM hierarchy allows us to evaluate our method in terms of the number of key concepts identified when partitioning the model. The idea is that the root of each subtree distinguished by our partitioning algorithm should denote a unique subfield of computer science. In order to determine such subfields that should be identified by the method, we analyzed the organization of computer science departments of Dutch universities with respect to the topics they used to identify subdivisions of their department. We then manually aligned these topics with the ACM topic hierarchy by translating the topic found into terms appearing in the ACM topic hierarchy. In cases where the topic matched more than one ACM terms (e.g. databases and information systems) both terms were counted. Terms that do not have a counterpart in the ACM hierarchy were ignored (e.g. 'mediamatics').

The test set consisted of 13 Dutch universities. Ten out of these had computer science departments. We extracted 85 terms from the corresponding web sites, mostly names of departments, groups or institutes. We were able to map 77 of these terms into 42 distinct terms from the ACM hierarchy. We distinguish three subsets of these 42 terms: terms that occur at least once, terms that occur at least twice and terms that occur at least three times. We can assume that terms that occur more than once to be important subfields of computer science that we would like to capture in a single module.

We compared these extracted terms with the root concepts of subtrees of the ACM hierarchy generated using our partitioning method. We chose to use a setting where the maximal size of an island is set to 100 and the threshold for merging islands is 0.2. With these settings, the method generated 23 modules. We decided to ignore three of the root terms:

ACM CS Classification This is the root of the hierarchy and not a proper term denoting a computer science topic

Mathematics of Computation The subtopics of this will normally be found in mathematics rather than computer science departments and were therefore not covered by our test set.

Hardware The subtopics of this module will normally be found in electrical engineering rather than computer science departments.

After this normalization, we compared the root terms of the generated modules with the terms identified on the department web pages and used overlap to compute the quality of the partitioning in terms of precision and recall of our method.

4.2 Results

We ran our method on the ACM classification using the hierarchy as the dependency graph. We further set the upper limit for the size of an island to 100 and the threshold value for merging islands to 0.2. The result are 23 subtrees with the following root nodes:

1. Numerical Analysis
2. Image Processing and Computer Vision
3. Management of Computing and Information Systems
4. Computing Milieux
5. Software Engineering
6. Computer Communication Networks
7. Data
8. Information Storage and Retrieval
9. Operating Systems
10. Database Management
11. Computer Systems Organization
12. Information Interfaces and Presentation
13. Software
14. (Mathematics of Computing)
15. Theory of Computation
16. (ACM CS Classification)
17. Information Systems
18. Computer Applications
19. Simulation and Modeling
20. Artificial Intelligence
21. Computer Graphics
22. Computing Methodologies
23. (Hardware)

As mentioned above, we disregarded the three nodes Mathematics of Computing, ACM CS Classification and Hardware. As a result, we receive 20 terms that our method determined to represent important subareas of computer science.

From the web pages of Dutch computer science departments, we extracted the 42 ACM terms shown in table 2. The most often occurring term was 'Algorithms' that described 5 groups, followed by 'Software' and 'Software Engineering'. Other frequently appearing topics were 'Robotics, 'Computer Systems', 'Computer Graphics', Information Systems', 'Expert Systems and Applications' (often referred to as 'Intelligent Systems'), Life Science applications, 'Systems Theory' and 'Theory of Computation'.

Test Set	Precision		Recall		F-Measure
> 2	30%	6 of 20	54.55%	6 of 11	38.71%
> 1	40%	8 of 20	42.11%	8 of 19	41.03%
> 0	60%	12 of 20	28.57%	12 of 42	38.71%

Table 1. Summary of evaluation results

We can see that there is quite some overlap between the root nodes of the subtrees determined by our methods and the terms from the test set. The overlap is especially striking when we only consider the set of terms that occurred more than two times in the description of groups. Six out of these eleven terms where also determined by our method. The recall becomes worse when considering terms than only occurred twice or once. This was expected, however, because there are single research groups on more specific topics such as distributed databases that are not necessarily regarded as important subfields by a large majority of people. We included these terms with less support in the test set to evaluate how many of the terms found by our method are used to describe the topics of groups. It turns out that 12 out of the 20 terms occur in the test set leading to a maximal precision of 60% for the largest test set. We used to F-Measure $((2 * (precision * recall))/(precision + recall))$ to determine the overall quality of the results. It turns out that we receive the best results on the set of terms that occur at least twice. A summary of the results is shown in table 1.

5 Discussion

We presented a method for automatically partitioning concept hierarchies using methods from social network analysis. We discussed the use of such methods for determining the strength of dependencies between classes, for determining sets of strongly related concepts and as a basis for a new heuristic for improving the result of the partitioning process. Further, we evaluate the method on the ACM classification of computer science topics by comparing the partitioning result with information about important computer science topics extracted from the web sites of (all) Dutch Computer Science Departments.

The main observation is that there is a significant overlap between topics that occur in the name of computer science research groups and the root nodes of the subtrees determined by our method. We were able to reach a precision of up to 60 percent when considering all terms occurring on the web sites. When only considering terms that are used more than two times, our method reached a recall of almost 55 percent. This can be considered a very good result as the chance of picking the most frequently occurring terms from the ACM hierarchy is $\binom{11}{1300}$ (the binomial of 11 over 1300) and we do not have more information than the pure structure of the concept hierarchy.

This result supports our claim, that the structure of concept hierarchies contains important information about key concepts that in turn can be used to partition the

occ.	ACM term
> 2	Algorithms
	Software
	Software Engineering
	Robotics
	Computer Systems Organization
	Computer Graphics
	Information Systems
	Applications And Expert Systems
	Life And Medical Sciences
	Systems Theory
	Theory Of Computation
> 1	User Interfaces
	Programming Techniques
	Artificial Augmented And Virtual Realities
	Artificial Intelligence
	Image Processing And Computer Vision
	Input/Output And Data Communications
	Parallelism And Concurrency
	Probability And Statistics
> 0	Computer-Communication Networks
	Business
	Computing Methodologies
	Control Design
	Decision Support
	Distributed Artificial Intelligence
	Distributed Databases
	Formal Methods
	Games
	Information Search And Retrieval
	Information Theory
	Management Of Computing And Information Systems
	Microcomputers
	Natural Language Processing
	Neural Nets
	Numerical Analysis
	Physical Sciences And Engineering
	Real-Time And Embedded Systems
	Security
	Signal Processing
	Software Development
	System Architectures
	Systems Analysis And Design

Table 2. ACM terms extracted from web sites of Dutch Computer Science Departments

hierarchy. Our hypothesis is, that this phenomenon is not random, but that people, when creating classification hierarchies are more careful when determining the subclasses of important classes. The result is a high number of children that cause our method to split the hierarchy at this particular point.

Of course the test set we used in our experiment is biased towards the particular strengthes of the Dutch Computer Science Community and does not necessarily reflect a neutral view on the importance of topics. We will try to overcome this problem in future work by repeating the experiment with computer science departments of a different country. Further, we plan to use human subject testing to support the current evaluation. We are currently preparing an experiment where people are asked to pick terms from the list of classes in the ACM hierarchy that they consider to represent important subfields of computer science. This test will be carried out in the computer science department of the Vrije Universiteit. As the members of the department have quite a number of different nationalities we hope to reduce the bias towards a particular nationality.

References

1. E. Amir and S. McIlraith. Partition-based logical reasoning for first-order and propositional theories. *Artificial Intelligence*, 2005. Accepted for Publication.
2. V. Batagelj. Analysis of large networks - islands. Presented at Dagstuhl seminar 03361: Algorithmic Aspects of Large and Complex Networks, August/September 2003.
3. R.S. Burt. *Structural Holes. The Social Structure of Competition.* Harvard University Press, 1992.
4. Y. Kalfoglou. Using ontologies to support and critique decisions. In *Proceedings of the 1st International Conference on Knowledge Engineering and Decision Support (ICKEDS'04)*, Oporto, Portugal, July 2004.
5. Peter Mika. Flink: Using semantic web technology for the presentation and analysis of online social networks. *Journal of Web Semantics*, 2005. to appear.
6. A. Rector. Modularisation of domain ontologies implemented in description logics and related formalisms including OWL. In *Proceedings of the 16th International FLAIRS Conference*. AAAI, 2003.
7. Heiner Stuckenschmidt and Michel Klein. Structure-based partitioning of large concept hierarchies. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *Proceedings of the Third International Semantic Web Conference (ISWC 2004)*, pages 289–303, Hiroshima, Japan, nov 2004.

BuddyFinder-CORDER: Leveraging Social Networks for Matchmaking by Opportunistic Discovery

Jianhan Zhu[a], Marc Eisenstadt[a], Alexandre Gonçalves[b], Chris Denham[a]

[a]Knowledge Media Institute, The Open University
{j.zhu, m.eisenstadt, c.m.denham}@open.ac.uk
[b]Stela Group, Federal University of Santa Catarina
a.l.goncalves@stela.org.br

Abstract. Online social networking tools are extremely popular, but can miss potential discoveries latent in the social 'fabric'. Matchmaking services can do naive profile matching with old database technology, and modern ontological markup, though powerful, can be onerous at data-input time. In this paper, we present a system called BuddyFinder-CORDER which can automatically produce a ranked list of buddies to match a user's search requirements specified in a term-based query, even in the absence of stored user-profiles. We integrate an online social networking search tool called BuddyFinder with a text mining method called CORDER to rank a list of online users based on 'inferred profiles' of these users in the form of scavenged Web pages.

1 Introduction and Motivation

Online social networking tools and services, in the form of friendship networks, instant messaging and chatting tools, dating services and business partner tools are extremely popular on the Web today – an extensive and evolving survey/taxonomy of Social Networking Services is provided online in [16]. Such tools and services have evolved from early-adopter 'leisure' tools to mission-critical business collaboration tools, and have attracted increasing attention from the research community [17] [18] [6] [7]. Today's online social networking tools typically involve a large number of users who coalesce into a community of mixed backgrounds and preferences, exhibiting varying degrees of overlapping interests and social cohesion. An online user typically has a number of contacts or buddies in his/her interest groups, with the amount of overlap (shared interests) dropping off dramatically as degrees of separation increase: groups may include work colleagues, family members, friends, conference acquaintances, friends-of-friends, recommended contacts, deliberately sought-out contacts, and of course random or even unwanted contacts. Current social networking tools allow users to manage additions to and deletions from their buddy lists manually, although most allow import from standard productivity-tool address books and preferences to block unwanted contacts.

In this era of powerful search engines (which are getting even better as the semantic web matures), consider the problem of seeking mission-critical or immediate advice

characterized by the question 'Who is available who can help me deal with an urgent problem *now*?' The Knowledge Management mantra of 'the right knowledge in the right place at the right time' rings somewhat hollow if you need to find key people quickly but cannot! Mixed hi-tech and lo-tech solutions (e.g. using Google or a social networking service as a first pass, then emailing or phoning around) may miss out on key availability information, which is one of the great strengths of instant messaging (IM) tools. Indeed, in this context it is not the messaging and chat capabilities of IM that pay the biggest dividends, but rather the *presence* information that (when correct at least!) shows who is available [19]. Modern social networking services such as Tribe [20], and indeed most discussion forums, typically include a 'presence indicator icon' so users can see at a glance who is available. Other tools, such as our own BuddySpace [21], also include the option of overlaying presence indicators on top of custom maps to deal with those cases where location is either important or simply 'feels good' to the user.

How, then, can we address the problem of finding the right person *now*? This problem falls at the overlap of two seemingly contradictory niches:

- *Good availability, but no need for search:* this is the 'IM niche' which is great for seeing who is available now, but since a user's IM buddy list is populated by people they already know, it may seem rather artificial to have to search through such people for matches
- *Web-centric search, but poor availability info*: the generic problem of searching the web for 'the right person' is well known, and indeed addressed by other papers at this workshop, but generally we cannot count on availability information being provided

We address this contradiction by noting a key exception to the assumption that 'a user's IM buddy list is populated by people they already know'. In particular, enterprise-wide IM and any service that provides automatic buddy-list population ('automatic roster or contact list generation'), has the potential for auto-enrolling users into large groups, such as student cohort groups and large just-in-time project teams, where a hierarchically-grouped buddy lists can grow just beyond a 'familiarity horizon': in other words, a user does *not* know a lot about everyone on the buddy list, yet it contains, within acceptable privacy limits, availability information for those who may have the knowledge to help solve a user's problem! A typical case is our own ELeGI project [22], a 23-partner European consortium with several hundred individuals involved. All are automatically subscribed to the IM roster, yet not everyone generally knows much about everyone else. Geographical information is also provided on our user maps, which can be overlaid with presence information, as shown in Fig. 1.

We believe that such usage scenarios will be increasingly typical of future knowledge workers and e-learners. In these cases, it is clear that simply listing online users in alphabetical order [1] [3] and randomly matching [4] cannot help users find their those with the right knowledge accurately and efficiently. Of course, some tools use registration information in profiles of users for search and can provide accurate search results given that these data are accurate and complete. This is the classic approach of the online dating services, and can be very effective in a live instant messaging context too. In Fig. 2, we see how MSN member search [2] enables users to search other users by their MSN nickname, last name, first name, and gender etc.

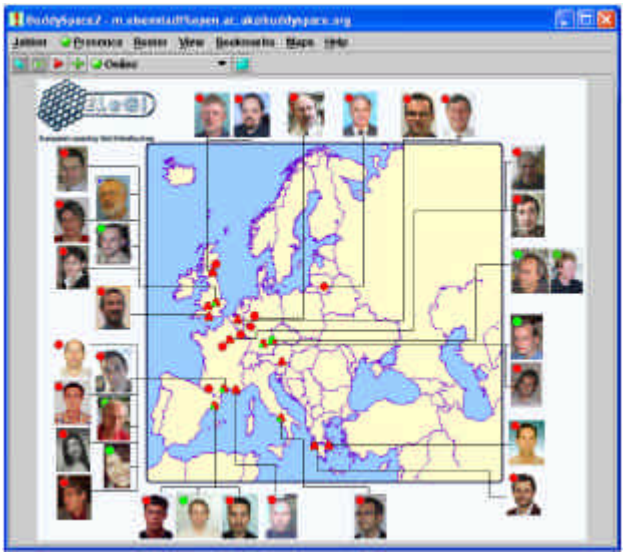

Fig. 1. BuddySpace showing presence (colored dots) and location information.

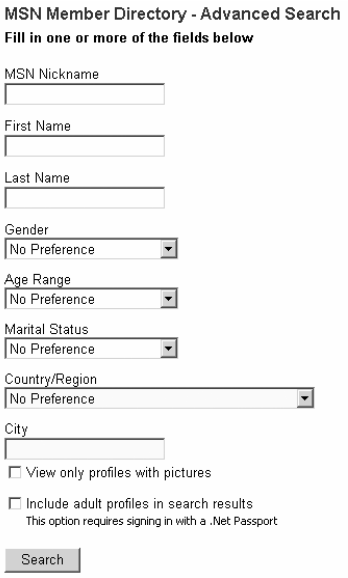

Fig. 2. MSN advanced search.

A more powerful approach in principle is to ask online users to write FOAF (Friend of a friend) files to describe themselves [23]. We could then perform buddy search using structured information in these FOAF files. However, these profile data or

FOAF files generally need to be explicitly and voluntarily provided by the users. Overall, we note four shortcomings of the database-profile and FOAF-profile approaches as follows.

- First, credibility of the data is at danger, i.e., users may not have provided true information about themselves or may not provide complete information about themselves, and thus the approaches is susceptible to abuse.
- Second, asking users manually to provide the data creates unacceptable overheads.
- Third, completeness of the data is at risk, i.e., users may not update the data to reflect their ongoing activities.
- Fourth, the search is limited to the information specified in the profile or FOAF file, thus the search cannot extend to a wider range of searches.

Another approach is to use a domain ontology for community of practice (COP) discovery within an organization. ONTOCOPI (Ontology-based Community of Practice Identifier) [18] attempts to uncover COPs by applying a set of ontology-based network analysis techniques that examine the connectivity of instances in the knowledge base with respect to the type, density, and weight of these connections. These COPs can be used for buddy search within an organization. Although effective, this approach also suffers from the following two shortcomings.

- First, it is generally hard to maintain the ontology to reflect the reality on the ground, i.e., part of knowledge in the ontology will become out of date quickly and new knowledge in reality cannot be put into the ontology in time.
- Second, the search is limited to knowledge in the ontology.

In contrast to these approaches, consider that there is significant information about many Web users in their personal homepages and blogs, and (even when those are missing, incomplete, or out of date) their friends or colleagues' homepages, and Web sites of their work place – all potentially relevant to their personal and/or career life. The information can provide a fertile ground for a social networking tool to search for buddies who match a user's preferences, *even in the absence of a user's specific profile information*, and *even in the absence of a user's own web pages*! Our belief is that we can leverage social networks ('whom you know') for opportunistic discovery that can potentially deliver the right knowledge in the right place at the right time, by deploying a mixture of buddy list, presence, and text mining methods. Such a hybrid approach can overcome the shortcomings of the current approaches.

- First, since Web pages about Web users are from various sources, these sources are controlled by different authorities and thus the search is less susceptible to abuse. We can judge the credibility of the data source for including the Web pages in search process.
- Second, there are fewer overheads in collecting the data. Users do not need to fill a form or maintain a FOAF file, and only need to make no or very little effort, e.g., list URLs of pages which can serve as profiles of themselves.
- Third, we can more easily get a more complete profile of a user and are not limited to information in a registration form or a FOAF file.
- Fourth, there are a more variety of searches that can be issued and the search is not limited to information in a registration form or a FOAF file.

In order to use Web pages pertinent to social networking users for searching them, we can directly use current search engines such as content-based search engines [5] and Google which uses both content information and PageRanks [8] of Web pages. PageRanks of Web pages are based on link structures of the Web. For example, given a list of users and their names, each of them has a profile consisting of a list of Web pages. In content-based buddy search, an online user can specify a query as his/her preferences or matching requirements. The query consists of a list of terms and Boolean relations between these terms. The query is used to search for other users as buddies whose profiles contain Web pages matching the query. These buddies are seen as matching the preferences or requirements of the user. For example, a query "Java AND C++" will return users whose profiles contain Web pages matching both "Java" and "C++" keywords. These users are assumed to have interests in "Java" and "C++".

However, a typical query such as "Java AND C++" will return a larger number of matching buddies, who are of different levels of relevance to the query. By giving a relevance score to each of these buddies, we can present a user with a list of buddies ranked by their relevance scores. In the ranked list of buddies, buddies more relevant to the query are on the top and buddies who are probably false positives are at the bottom and can be ignored. We observe that a buddy is more relevant to a term in a query when the buddy's name has more co-occurrences with the term and/or occur close to the term in co-occurred Web pages. In content-based search, we can rank buddies by only taking into account co-occurrences between each buddy and terms in the query.

In our previous work, we have proposed a text mining method called CORDER [9], which unearths relations between named entities[1] from Web pages of a community. CORDER is based on communities of practice [10], where a group of people are collaborating together on shared tasks, rather than their institutional division. The documents that a group of people produces mirror what people do and who they work with. CORDER automatically discovers relations between people in the community from these documents. Given a named entity, both co-occurrences and distances between the named entity and other named entities in these documents are taken into account in ranking these named entities. Our experimental results showed that CORDER produced better rankings of named entities than the co-occurrence based ranking method.

In this paper, we apply the CORDER method to buddy search. There are mainly two extensions of the CORDER method in the buddy search scenario. First, in previous CORDER method, we consider relations between named entities. While in buddy search, we consider relations between a named entity and terms in a query. While our previous experiments show that CORDER can identify significant relations between named entities. In buddy search, we need to find out whether CORDER can also identify significant relations between named entities and terms. Second, we need to extend CORDER method to take into account Boolean relations between terms in a query.

Given a query, the ranking of a user is given by taking into account the co-occurrences between the user and terms in the query, the distances between the user and terms in the query, and Boolean relations between the terms. In an online group

[1] Named entities are proper names of various types, e.g., "John Smith" is a "Person" and "Open University" is an "Organization".

consisting of a number of buddies, Web pages in one buddy's profile often talk about not only him/herself but also his/her friends, i.e., other buddies, in the same group. One buddy may belong to various groups on the Web. In buddy ranking, given a buddy, Web pages from both his/her profile and profiles of other buddies in the same groups as his/hers are taken into account in ranking him/her. We integrate CORDER with an online social networking tool called BuddyFinder. BuddyFinder is part of BuddySpace (http://buddyspace.sourceforge.net/) [11], an online instant messaging tool. A number of users are subscribed to BuddyFinder and each of them *has an optional* profile containing a list of Web pages from his/her homepage, blog, or other sources describing his/her personal or career life, and even in the absence of that profile we can make a best guess based on the user's identity and domain (part of their login specification) to find such pages. A BuddyFinder user can use term-based query to search subscribed users and get a list of buddies ranked by CORDER. In Fig. 3, a BuddyFinder user input a query "semantic web" OR ontology and get a list of users ranked by CORDER. He/she can choose to interact with these users by chatting, emails, and viewing their profiles etc.

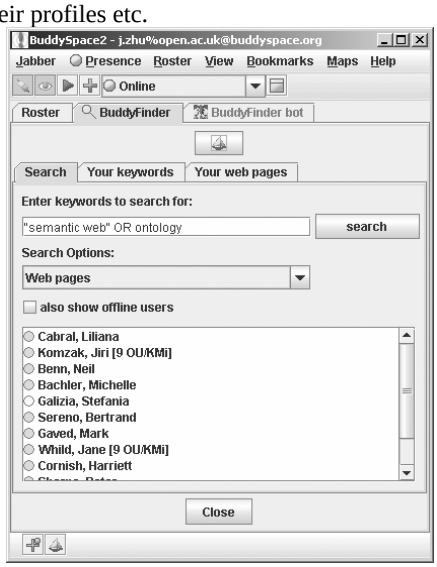

Fig. 3. BuddyFinder output for a search on "semantic web" OR ontology.

The rest of the paper is organized as follows. In Section 2, we give an overview of the CORDER method. In section 3, we present the BuddyFinder-CORDER system. In Section 4, we conclude and propose future work.

2 CORDER: A Community Relation Discovery Method

Typical questions for knowledge managers are what do your employees know about, which of your customers have they contacts with, and who works well together in

teams? However, the knowledge represented in organizational ontologies and other resources is often static, reflecting management's design of what should happen in the organization and not necessarily the real situation on the ground. The real situation is often characterized better by communities of practice [10], the groupings of people who collaborate together on shared tasks, rather than institutional divisions.

In our buddy search scenario, online users as a community have web pages in their profiles describing themselves and each other. These web pages can often more truthfully describe themselves than their registration information or FOAF files, which are often static and not up to date.

We argue that Web pages as profiles of online users mirror what these users do and who they share interests with. CORDER discovers relations from the Web pages of a community of online users. Previous CORDER method is based on co-occurrences of named entities (NEs) and the distances between them. A named entity recognizer, ESpotter [14], is used to recognize people, projects, organizations and research areas from Web pages. BuddyFinder extends the CORDER method to relations between buddy names as NEs and terms in a Boolean query.

In the CORDER method, given a target as an NE, we rank a number of co-occurring NEs as objects. In buddy search scenario, the target is a term in a Boolean query and the term co-occurs with a number of buddy names as objects. We assume that objects that are closely related to the target tend to appear more often with the target and closer to the target in Web pages. Given the target, we calculate a relation strength for each co-occurring object based on its co-occurrences and distances from the target. The co-occurring objects are ranked by their relation strengths. Thus objects which have strong relations with the target can be identified. The relation strength between a target and an object takes into account three aspects as follows.

Co-occurrence. A target and an object are considered to co-occur if they appear in the same Web page. Generally, if the object is closely related to the target, they tend to co-occur more often.

Distance. A target and an object which are closely related tend to occur close to each other.

Frequency. A target or object is considered to be more important if it has more occurrences in a Web page.

Given a target, the higher the relation strength of an object, the closer they are related to each other. We set a relation strength threshold, so that only significant relations having relation strengths above the threshold are selected. Relations having relation strengths below the threshold are considered to result from noise in our data and are ignored. In our study we set the threshold as the value at which a target and an object co-occur with only one occurrence each in only one Web page, and their distance in the Web page is a certain value *D*. Given the target, objects with their relation strengths above this threshold are considered to be related.

Evaluation of the CORDER method on a departmental Website indicates that the method can find NEs closely related to a target and provide accurate rankings. CORDER's running time increases linearly with the size and number of web pages it examines. CORDER can incrementally evaluate existing relations and discover new relations by taking into account new Web pages. Thus CORDER can scale well to a large dataset.

3 BuddyFinder-CORDER

In Section 3.1, we present the architecture of the BuddyFinder-CORDER system. In Section 3.2, we present the extended version of the CORDER method for buddy ranking. In Section 3.3, we present our initial user evaluation.

3.1 System Architecture

BuddyFinder-CORDER relies on an instant messaging platform Jabber [12] for exchanging information. In Fig. 4, Jabber [12] uses an XML based protocol called XMPP [12] for all transactions between Jabber users and the Jabber server in **B**. When a Jabber user in **A** issues a term-based query to search for buddies, the query is sent from the user's Jabber client using chat messages following the XMPP protocol. The BuddyFinder query is routed by the server to the BuddyFinder Chatbot in **C**, which interacts with various modules to get a ranked list of buddies. The BuddyFinder Chatbot is a standard Jabber client, an architecture which allows a lot of flexibility in terms of where it is hosted and allows BuddyFinder users to interact with it using any Jabber based instant messaging client software. BuddyFinder Chatbot will reply with the results of the query following the XMPP protocol.

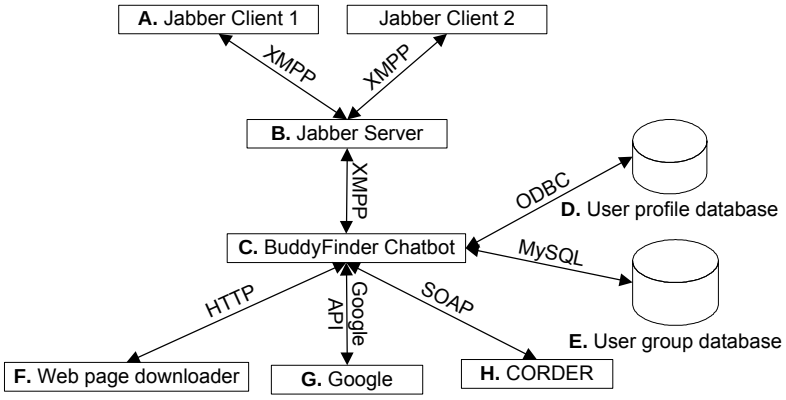

Fig. 4. BuddyFinder-CORDER Architecture.

The user profile database in **D** stores each user's profile consisting of a list of Web page URLs. BuddyFinder Chatbot accesses user profile information via ODBC connection. The initial list of URLs for a user is generated automatically using a Web search component via the Google API. To give an easy start for a user to specify these URLs, the Web search component makes an "intelligent guess" in finding Web pages relevant to him/her. The intelligent guess is based on the domain shown in his/her email address and his/her name, which are stored in the user group database in **E**. For example, if one user's email address is m.eisenstadt@open.ac.uk and name is "Marc

Eisenstadt", we guess the domain of the user is open.ac.uk. We send the query "Marc Eisenstadt site:open.ac.uk" to Google and use URLs of the top 10 Web page as the default profile for the user.

To improve search performance, web pages in users' profiles are downloaded in **F** and cached for search. Each user is subscribed to various user groups, e.g., KMi user group and Open University user group. Given a user, a list of buddies who belong to his/her user groups is generated by issuing SQL queries on the profile database in **E**. When a user sends a query to search, BuddyFinder performs the search on the profiles of the list of buddies. Thus different users can get different search results even when they sent the same query.

When a user searches BuddyFinder, he/she may get a large number of buddies whose profiles all match the query. Some buddies who are not closely related to the query may also be returned simply because their names co-occur with terms in the query on Web pages. We need a ranking algorithm which can rank these buddies in terms of their relevance to the query. A good ranking algorithm should be able to both identify buddies highly relevant to the query by putting them at the top of the list and putting them in the correct order. Ideally, the ranking algorithm should put the buddies in the same order that the user will put them. In Fig. 4, given a search query, BuddyFinder uses the CORDER algorithm, which takes into account co-occurrences, distances in co-occurred Web pages, and relations in the query, to rank search results.

The buddy search process in Fig. 4 is as follows. A user in **A** specifies a search query in a command line, e.g., "find "semantic web" OR ontology". The query is sent to BuddyFinder Chatbot in **C** and be interpreted for a list of terms and Boolean relations between them. The BuddyFinder Chatbot goes to the user group database in **E** to find a list of buddies who belong to the same groups as the current user. BuddyFinder Chatbot goes to user profile database in **D** to get profiles of these buddies.

Each buddy's profile consists of a list of Web pages. Given a buddy, BuddyFinder Chatbot uses a set of Web pages for ranking him/her against the query. First, Web pages from the buddy's profile are included in ranking. Second, since Web pages from profiles of other buddies who are in the same groups as his/hers, such as the buddy's colleagues, contain information relevant to him/her, e.g., co-authoring same papers and members of same projects, these Web pages are also taken into account for ranking him/her.

Given the buddy, for each term in the query, BuddyFinder Chatbot processes the set of Web pages for co-occurrences between the buddy's name and the term and offsets of the buddy's name and the term in co-occurred Web pages. CORDER in **H** is implemented as a Web service and BuddyFinder Chatbot calls the CORDER Web service by sending co-occurrence, offset and Boolean relations information to CORDER as a SOAP message. CORDER calculates a ranking of the buddy against the query based on the algorithm presented in the next section and sends back the ranking to BuddyFinder Chatbot as a SOAP message. BuddyFinder presents the buddies in the order of their rankings to the user.

3.2 Buddy Ranking Algorithm

We extend the CORDER method in mainly two aspects to the buddy search scenario. First, extend relations between named entities to relations between named entities and terms. Second, extend CORDER method by taking into account Boolean relations between terms.

An online user's preferences for buddies are specified in a term-based query, which consists of a list of terms and the Boolean relations between these terms. These terms can either be single words or phrases. We use the CORDER method to calculate the ranking of a buddy as a relation strength between the buddy and the search query. The relation strength is based on co-occurrences of the buddy's name and each term in Web pages, the distances between the buddy's name and each term in Web pages, and Boolean relations between these terms. A list of buddies is ranked by their relation strengths, and buddies who are more relevant to the query are at the top of the list. Given a buddy, Web pages from his/her profile and profiles of other buddies in the same groups as his/hers are used for ranking him/her. The relation strength between a buddy and the query takes into account four aspects as follows.

Co-occurrence of buddy and a term. A buddy and a term are considered to co-occur if they appear in the same Web page. Generally, if a buddy is closely related to a term, they tend to co-occur more often. For a buddy, *B* and a term *K*, we use Resnik [13]'s method to compute a relative frequency of co-occurrences of *B* and *K* as $\hat{p}(B,K) = \frac{Num(B,K)}{N}$, where *Num(B,K)* is the number of co-occurring Web pages for *B* and *K*, and *N* is the total number of Web pages.

Distance between buddy's name and a term. A buddy and a term which are closely related tend to occur close to each other. If a buddy and a term, *B* and *K*, both occur only once in a Web page, the distance between *B* and *K* is the difference between the offsets of *B* and *K*. If *B* occurs once and *K* occurs multiple times in the Web page, the distance of *B* from *K* is the difference between the offset of *B* and the offset of the closest occurrence of *K*. When both *B* and *K* occur multiple times in the Web page, we average the distance from each occurrence of *B* to *K* and define the logarithm distance between *B* and *K* in the *i*th Web page as $\overline{d_i}(B,K) = \frac{\sum_j (1+\log_2(\min(B_j,K)))}{Freq_i(B)}$, where $Freq_i(B)$ is the number of occurrences of *B* in the *i*th Web page and $\min(B_j,K)$ is the distance between the *j*th occurrence of *B*, B_j, and *K*.

Frequency of buddy's name. A buddy is considered to be more important if his/her name has more occurrences in a Web page. Consequently, a more important buddy on a Web page tends to have strong relations with other terms which also occur on the Web page.

Boolean relation between terms. Terms are connected by AND, OR, NOT Boolean connectors. There Boolean connectors are taken into account in relation strength calculation. For two terms *T1* and *T2*, we consider two kinds of relations between them as AND and OR. The AND operator is used to specify that both *T1* and *T2* must evaluate to TRUE for "*T1* AND *T2*" to evaluate to TRUE. Given a buddy, *B*, we use a set of Web pages to rank him/her. In order for both *T1* and *T2* to evaluate to TRUE, *B* needs to co-occur with *T1* and *T2*, respectively. For example, if *B* co-occurs with both *T1* and *T2* on a page, "*T1* AND *T2*" evaluates to TRUE. If *B* co-occurs with *T1* on one page and co-occurs with *T2* on another page, "*T1* AND *T2*" still evaluates to TRUE. If the relation strengths between *T1*, *T2* and a buddy are $R(B,T1)$ and $R(B,T2)$, respectively, we define the relation strength between "*T1* AND *T2*" and the buddy as $R(B,T1 \ \ AND \ \ T2) = R(B,T1) \times R(B,T2)$. The OR operator is used to specify that either *T1* or *T2* must evaluate to TRUE for "*T1* OR *T2*" to evaluate to TRUE. In order for either *T1* or *T2* to evaluate to TRUE, *B* needs to co-occur with either *T1* or *T2*. We define the relation strength between "*T1* OR *T2*" and the buddy as $R(B,T1 \ \ OR \ \ T2) = R(B,T1) + R(B,T2)$. For one term *T1*, the NOT operator is used to specify that *T1* must evaluate to FALSE for "NOT *T1*" to evaluate to TRUE. We define the relation strength between "NOT *T1*" and the buddy as $R(B, NOT \ \ T1) = 0$ if $R(B,T1) > 0$ and =1 if $R(B,T1) = 0$.

Relation strength between buddy and query. Given a buddy, *B*, and a query, *Q*, we get a list of terms $\{T_k\}$ from *Q* and their Boolean relations. We calculate the relation strength between *B* and *Q* by taking into account co-occurrences, distance and frequency between *B* and each term in $\{T_k\}$, and Boolean relations between terms in $\{T_k\}$. The relation strength, $R(B,T_k)$, between *B* and T_k is defined in Equation 1.

$$R(B,T_k) = \hat{p}(B,T_k) \times \sum_i \left(\frac{f(Freq_i(B)) \times f(Freq_i(T_k))}{\overline{d}_i(B,T_k)} \right) \qquad (1)$$

where $f(Freq_i(B)) = 1 + \log_2(Freq_i(B))$, $f(Freq_i(T_k)) = 1 + \log_2(Freq_i(T_k))$, $Freq_i(B)$ and $Freq_i(T_k)$ are the numbers of occurrences of *B* and T_k in the *i*th Web page respectively.

We get the relation strength, $R(B,Q)$, between *B* and *Q*, by combining $R(B,T_k)$ using the Boolean relations between them. For example, for a query *Q* = T1 AND T2 AND (NOT T3), $R(B,Q) = R(B,T1) \times R(B,T2) \times R(B, NOT \ \ T1)$.

3.3 Initial User Evaluation

Currently we have 234 BuddyFinder users, each of them can have a profile consisting of up to 10 Web pages from their homepages, departmental Web pages, and blogs etc. We have asked three BuddyFinder users from outside our department to independ-

ently evaluate our system. Each of them was asked to compose 10 queries and used them to search in BuddyFinder. We compare CORDER with co-occurrence based ranking method, in which we calculate the relation strength between a buddy and a query term as the number of co-occurrences of the two in Web pages.

For each query, the user got two ranked lists of buddies and he/she was not told which one was created by CORDER or the co-occurrence method. The user was asked to judge the relevance of each of the top 5 buddies in the two lists to the search query by giving a score from -2 to 2, where -2 is highly irrelevant, -1 is irrelevant, 0 is not sure, 1 is relevant, and 2 is highly relevant. The user can judge the relevance by chatting with the buddy and viewing his/her profile etc.

For the first user, we got a total score of 89 and 78 (out of 100) on the 10 queries for the CORDER based rankings and co-occurrence based rankings, respectively. For the second user, we got a total score of 86 and 71 (out of 100) on the 10 queries for the CORDER based rankings and co-occurrence based rankings, respectively. For the third user, we got a total score of 85 and 78 (out of 100) on the 10 queries for the CORDER based rankings and co-occurrence based rankings, respectively.

The initial results show that CORDER produced good rankings for buddy search and provided better rankings than the co-occurrence based method. We are currently working on evaluating BuddyFinder-CORDER on a larger user group and the initial results are promising.

4 Conclusions and Future Work

We have shown that the BuddyFinder-CORDER system can help users' online collaboration by enabling them to search for buddies matching their interests specified in term-based queries. Since current instant messaging tools mostly rely on registration information for buddy search, they are susceptible to fraud, limited information, and out-of-date information. BuddyFinder-CORDER can enable more trustworthy, more versatile, and more up-to-date buddy search. Initial experiments show that BuddyFinder-CORDER can find buddies highly relevant to search queries and provide better rankings than a co-occurrence based method. BuddyFinder-CORDER's running time increases linearly with the size and number of Web pages it examines. Thus BuddyFinder-CORDER can scale well to a large dataset. The initial experiment is still preliminary. We are evaluating BuddyFinder-CORDER on a larger user group consisting of over 200 users.

CORDER's rankings are derived from data mined from a collection of documents. In this way it gives a wider view of the "world" of a domain than data from a single source, such as registration information provided by users. Our future work is two-folded. First, BuddyFinder-CORDER uses a text mining method to deal with mainly unstructured information in Web pages. If we can get structured information about online users in the form of registration data or FOAF files, we can improve the BuddyFinder-CORDER method by taking into account both unstructured and structured data. For organizational use, we are considering using CORDER text mining methods to keep an organizational ontology up to date with knowledge embedded in documents

produced by the organization. The ontology could be used by systems such as ONTOCOPI [18] for community of practice discovery and buddy search. Second, BuddyFinder-CORDER enables discovery of indirect relationships among buddies. In our scenario, it can be used to find out buddies who are not directly related to terms in a query but are interesting, e.g., a buddy A is highly relevant to a buddy B, B is highly relevant to a query, A is not directly relevant to the query. There is an indirect relationship between A and the query. BuddyFinder can suggest A to the searcher. We are experimenting with using the closest entities suggested by CORDER to improve the vector descriptions of documents for clustering. Our initial experiments suggest that this approach produces clusters which score as well as the widely used SOM method [16] on a total information gain measure of cluster quality. The execution time of the CORDER enhanced clustering method however increases linearly with the size and number of documents it examines so that it starts to outperform SOM on collections of more than 700 vectors.

Acknowledgements

Work on BuddySpace and BuddyFinder is supported by the European Community under the Information Society Technologies (IST) programme of the 6th Framework Programme for RTD – project ELeGI, contract IST-002205. This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of data appearing therein.

Work on CORDER was partially supported by the Designing Adaptive Information Extraction from Text for Knowledge Management (Dot.Kom) project, Framework V, under grant IST-2001-34038 and the Advanced Knowledge Technologies (AKT) project. AKT is an Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University. This research was also partially supported by the Brazilian National Research Council (CNPq) with a doctoral scholarship held by Alexandre Gonçalves.

We are grateful to Jiri Komzak for the implementation of BuddySpace and significant input into BuddyFinder, Martin Dzbor for significant input into the concepts, design, and implementations underlying BuddySpace, and Victoria Uren and Enrico Motta for helpful discussion.

References

1. http://www.msn.com
2. http://members.msn.com
3. http://messenger.yahoo.com
4. http://www.icq.com
5. http://www.verity.com

6. Isaacs, E., Walendowski, A., Whittaker, S., Schiano, D. J., Kamm, C. A.: The character, functions, and styles of instant messaging in the workplace. In Proc. of ACM conference on Computer supported cooperative work (CSCW 2002), 11-20.
7. Setlock, L. D., Fussell, S. R., Neuwirth, C.: Taking it out of context: collaborating within and across cultures in face-to-face settings and via instant messaging. In Proc. of ACM conference on Computer supported cooperative work (CSCW 2004), 604-613.
8. Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks 30(1-7): 107-117 (1998)
9. Zhu, J., Gonçalves, A. L., Uren, V. S., Motta, E., Pacheco, R.: Mining Web Data for Competency Management. In Proc. of 2005 IEEE/WIC/ACM International Conference on Web Intelligence 2005 (WI'2005), Compiegne University of Technology, France, September 19-22.
10. Wenger, E.: Communities of Practice, learning meaning and identity. Cambridge University Press (1998).
11. Eisenstadt, M., Komzak, J. and Dzbor, M.: Instant messaging + maps = powerful collaboration tools for distance learning. In Proc. of TelEduc03, May 19-21, 2003, Havana, Cuba.
12. Adams, D. J.: Programming Jabber Extending XML Messaging. O'Reilly (2001).
13. Resnik, P.: Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. Journal of Artificial Intelligence Research (1999). 11:95-130.
14. Zhu, J., Uren, V. S., Motta, E.: ESpotter: Adaptive Named Entity Recognition for Web Browsing. In Proc. of 3rd Professional Knowledge Management Conference, Springer, LNAI (2005).
15. T. Kohonen, Self-Organizing Maps, Springer Series in Information Sciences, Vol. 30, 1995.
16. Meskill, J.: A Social Networking Services Meta-List: http://socialsoftware.weblogsinc.com/entry/9817137581524458/
17. Nardi, B.A., Whittaker, S., Isaacs, E., Creech, M., Johnson, J., and Hainsworth, J.: Integrating communication and information through ContactMap. *Communications of the ACM*, 45:4, 89-95 (2002).
18. Alani, H., S. Dasmahapatra, K. O'Hara, N. Shadbolt.: Identifying Communities of Practice through Ontology Network Analysis. IEEE Intelligent Systems 18(2): 18-25, (2003).
19. Chakraborty, R.: Presence: A Disruptive Technology, Presentation at JabberConf, Denver, (2001).
20. http://www.tribe.net
21. Vogiazou, I.T., Eisenstadt, M., Dzbor, M., and Komzak, J.: From Buddyspace to CitiTag: Large-scale Symbolic Presence for Community Building and Spontaneous Play. In Proc. of the ACM Symposium on Applied Computing, Santa Fe, New Mexico, March 13 -17, (2005).
22. Allison, C., Cerri, S.A., Gaeta, M., Ritrovato, P. and Salerno, S.: Human Learning as a Global Challenge: European Learning Grid Infrastructure. In Varis, T., Utsumi, T. and Klemm, W. (eds.), *Global Peace Through the Global University System*, RCVE, Tampere, 465-488, (2003), http://www.terena.nl/conferences/tnc2004/core_getfile.php?file_id=576
23. FOAF, http://www.foaf-project.org

The Social Semantics of LiveJournal FOAF: Structure and Change from 2004 to 2005

John C. Paolillo[1,2], Sarah Mercure[2] and Elijah Wright[2]

[1]School of Informatics, Indiana University
[2]School of Library and Information Science, Indiana University
Bloomington, Indiana, 47405 USA
{Paolillo, smercure, ellwrigh}@indiana.edu

Abstract. Social Network Analysis methods hold substantial promise for the analysis of Semantic Web metadata, but considerable work remains to be done to reconcile the methods and findings of Social Network Analysis with the data and inference methods of the Semantic Web. The present study develops a Social Network Analysis for the foaf:knows and foaf:interests relations of a sample of LiveJournal user profiles. The analysis demonstrates that although there are significant and generally stable structural regularities among both types of metadata, they are largely uncorrelated with each other. Also there are large local variations in the clusters obtained that mitigate their reliability for inference. Hence, while information useful for semantic inference over user profiles can be obtained in this way, the distributional nature of user profile data needs closer study.

1 Introduction

The Semantic Web is largely characterizable as an enterprise with one principal goal: the formalization and standardization of online metadata. Over the past few years, the purpose of this effort has been variously characterized as assisting interoperability of web information systems [1], facilitating Knowledge Management [2], and permitting the development of future web applications based on artificial intelligence techniques [3]. For this reason, it is somewhat surprising that social networking metadata, using the FOAF (Friend-of-a Friend) vocabulary, has emerged as possibly the single most prevalent use of Semantic Web technologies so far [4,5]. Numerous weblog and journal-hosting sites now export their user data using FOAF, alongside their content serialization in RSS.

On the surface, the two types of information are very different. For example, businesses needing to communicate about the stock, manufacturing conditions, materials and specifications of metal fasteners have very different communication needs from private individuals looking to maintain or develop a circle of friends and acquaintances. Moreover the word "ontology", used to describe a set of interlocking metadata definitions, suggests a static Platonic conception of objects in the world, whose existence is to be identified ("marked up") through the application of labels.

Social network information defies this sort of conception: is so-and-so really your "friend", or are you simply name dropping to enhance your appeal in the present conversation? Or are you more than friends, but happen to be downplaying your association with someone I may not approve of? Would two people agree on the meaning of friendship, even a shared friendship? And would this agreement be enough to permit other inferences to be drawn, as is the purpose of the Semantic Web? The parameters of social information are much harder to determine than Platonic semantics would allow, and therefore pose major challenges to the Semantic Web project.

The tools of Social Network Analysis (SNA) have developed to cope with this sort of issue. Typically, network analysts employ large amounts of readily-collected information — mutual naming patterns, event participation, helping relations, desirability ratings, etc. — to identify social positions, roles and relationships in aggregate, either through statistical means or by interpreting geometric representations of the information [6,7]. These methods offer a powerful window into social functions and processes from the most mundane (e.g. the pronunciation of English vowels by Detroit teenagers [8]) to the most rarified (e.g. US Supreme Court rulings, [9]). At the same time, observations obtained through network analysis are necessarily time-bound: they may not be true for any other time or context of observation. Hence, social network information must always be interpreted cautiously.

For SNA, the emergence of FOAF is a happy coincidence, in that it provides an inexpensive source of large amounts of reasonably rich social network data. The utility of FOAF for SNA is helped by its deliberate vagueness: relations such as foaf:knows are not explicitly tied to terms like "know" or "friend" in other ontologies like WordNet, and hence can be allowed to vary somewhat according to context. Furthermore, the availability of social network metadata makes it possible to consider the relationship of semantics to social structure, i.e. "emergent semantics", from the dynamic nature of social relations and their role in fostering and communicating semantic change. Many questions pertinent to the Semantic Web effort arise in this context. Do the ontologies we encode represent the semantic relations that people actually use? Can existing Semantic Web technologies encompass dynamic, context-bound meanings? How stable are these meanings over time? And to what extent can ontologies be adapted to such changing, context-bound meanings, to provide useful inferences? Hence, the Semantic Web also potentially benefits form the application of SNA methods, which might provide templates for evaluating the Semantic Web metadata in contexts where it is actually used.

It is this set of issues that motivates the current study, which examines the change in a set of LiveJournal profiles for friends (represented by foaf:knows) and interests (foaf:interests) of eighteen thousand users. By examining these data, we can potentially learn about the way in which people's online social spheres evolve alongside their changing interests. This provides a view onto the emergent semantic categories of interests and their relation to the social milieu.

2 Methods

For this study, we obtained two samples of LiveJournal profiles, collected one year apart. The first set of profiles consisted of a subset of the FOAF files collected by Jim Ley in March 2004 which happened to be LiveJournal profiles. Ley's purpose in obtaining the profiles was to provide a database of FOAF files upon which to develop some demonstration applications. The data were provided both in the form of raw RDF/XML files, and as a MySQL database of RDF triples. RDF statements pertaining to LiveJournal user accounts were readily identified through having common kinds of information (especially nick-names), and through their overall social coherence (the LiveJournal profile interface only permits users to friend other LiveJournal users). These users were identified and analyzed in depth in earlier work [5].

2.1 LiveJournal

LiveJournal is a service which permits users to create online personal journals. Created in 1999 by Brad Fitzpatrick as a way to keep in touch with his own friends, LiveJournal has since snowballed into a dynamic community of individual and group journals. In the spring of 2004 participation in LiveJournal reached nearly three million users and in just over one year's time since then, the number of users had ballooned to over seven million, according to the site's front page.

Users of LiveJournal include a wide range of individuals from all over the world. At this time most registered users are from the United States, Canada, the United Kingdom, the Russian Federation, and Australia, from most to least common. Of the users from the US, the most report that they live in California, with users from Florida, New York, Michigan, and Texas following closely behind. Over 60% of users are female, and the highest distribution of users fall within the 15-20 age range. Considering the broad range of user backgrounds we can only speculate as to the general reasons that users participate. Given the personal nature of data required for user registration, as well as the features provided by LiveJournal, it seems clear the site is intended for presenting personal journal-like information. Previous research supports this notion with evidence that journal-type blogs are the most common use of blogging tools [10]. Typical user pages contain personal accounts of the user's day or details of recent life events.

Upon registering for an account LiveJournal users are presented with a form which requests optional information from the user, such as birthdate, gender, geographic location, email address, etc. In addition the form provides space for the user to post an image and/or brief biography, set privacy controls for their content, and list interests. The interface for interests is a free-text box, so users are not at all limited in what or how many interests they declare. User information is exported as an automatically generated FOAF RDF/XML file [11], made available at a pre-determined URL. Users can update this information at any time.

Common interests listed by users are presented on a users "user info" page and hyperlinked to a list of all the Livejournal communities and users who list the same

interest, thereby facilitating the social engagement of users who share similar interests. The user info page also lists a user's list of "friends", who are typically other LiveJournal users. "Friending" another LiveJournal user allows one to do more than simply declare affiliation; it is also tantamount to entering a subscription to the content they produce. Users can take advantage of an automatically generated link on their main page that brings up the past several entries for each of their listed friends. The process of designating interests and friends thus creates a complex interlinked network of users, facilitating easy access to social groups and to people with common interests.

2.2 Data Handling

The RDF triples from the 2004 sample were imported into a PostgreSQL database, from which foaf:knows and the foaf:interest relations were extracted using database operations, along with other supporting relations, such as dc:title and foaf:nick, which allowed us to identify meaningful content with the relations. For foaf:interests, we identified the 500 most popular interests, and a subset of 21,506 user profiles mentioning these interests. For foaf:knows, we identified 500 most popular recipients of the foaf:knows relation, and 11,818 users' profiles stating that the associated person foaf:knows at least one of those people. Each relation was arranged into a binary incidence matrix, I_{2004} for foaf:interests and K_{2004} for foaf:knows, with cell values $I_{i,j}$ and $K_{i,j}$ being 1 or 0, indicating for each profile i whether a relation to a given interest or popular user j is present. These incidence matrices were used in the subsequent statistical analysis.

In June 2005, we took the same list of 21,506 LiveJournal users and retrieved current FOAF profiles for them by means of an automatic script. These were imported into SWI-Prolog, where Prolog rules were used in place of the database operations to obtain the foaf:interests and foaf:knows relations for the 500 most popular interests and users identified in the 2004 sample. These were arranged in a second pair of incidence matrices, I_{2005} and K_{2005}, as had been done for the 2004 profiles.

The four incidence matrices were carefully collated to produce a final incidence matrix for the complete profile data P, containing information from both the foaf:interests and foaf:knows relations, for both years. This resulted in a matrix of 18,725 rows and 2000 columns. FOAF profiles in the 2004 foaf:interests matrix that were in none of the three other matrices were dropped from consideration at this point. The foaf:knows relations showed the largest change over the two years, with nearly half of the users in each year not found in the other year's data. Such instances were handled by filling in the appropriate cells of P with zeros.

A key observation made at this point is that users' social relations — as represented by their orientation toward the most popular users — appear to fluctuate more than do their interests, but this appearance may be due to the greater sparsity of the foaf:knows relation, with respect to the foaf:interests relation.

2.3 Statistics

Statistical data handling was accomplished using R, the GPL statistical environment and programming language. Specifically, we conducted a Principal Components Analysis (PCA) of the matrix P, to identify the structure of inter-correlation among the interests and popular users (columns) of P. PCA is a statistical technique common in machine learning, and Latent Semantic Analysis [12] is a related technique. Co-citation analysis [13,14,15] is an alternative approach widely used in bibliometric studies which employs co-citation, a similar measure of relationship similar to correlation, which PCA is based on. Correlation has the advantage of being centered about the overall mean of the data, and scaled according to its overall variance. These treatments result in a projection of the data into a space whose greatest dimensions are the dimensions of greatest (co-)variation in the data itself, rather than an artificial set of dimensions representing an artifact of the analysis or observational procedures. Moreover, when distributional assumptions are met, the PCA supports recognized statistical inferences [16]. For these reasons, we felt that PCA would be an appropriate technique to allow us to compare the relations among different interests and popular users in our set of user profiles.

PCA of the matrix P is accomplished by Singular Value Decomposition (SVD) of a z-score standardized version of P, namely $Z = UdV^T$. The output matrices U and V are sets of eigenvectors representing the rows (U) and columns (V) of Z, and d is a diagonal matrix of singular values of Z. U and V are rectangular matrices with r columns, and d is an $r \times r$ square matrix, where r is the rank of matrix Z, or the number of interpretable principal components. Various methods are used to choose r, the most common being a scree plot, showing the rank-size relation among the singular values d. For interpretation, we use the factor scores of Z, where the column eigenvectors V are scaled using d: $F = Vd$. These vectors are treated as representing points in r-dimensional space, allowing us to measure their distances, for clustering the interests and popular users. Because of the size of P, R's built-in functions `prcomp()` and `princomp()` were unsuitable, and we wrote our own more lightweight wrappers for the function `svd()`, R's interface to the LAPACK routine for SVD.

A scree-plot was used to identify 20 dimensions of variation as potentially significant. These Principal Components were visualized as factor score plots. The factor scores were then split into four groups for foaf:knows and foaf:interest relations in 2004 and 2005, and these groups were submitted to hierarchical cluster analysis, using Euclidean distances and Ward's method. These cluster analyses were visualized as dendrograms, in order to identify the content and organizational structure of each cluster. Finally, the clusters were cross-tabulated for 2004 and 2005, and those cross-tabulations were visualized as networks, so that the dynamic re-organization of the foaf:knows and foaf:interest relations could be studied. The results of these analyses are presented below.

3 Results

The first set of results concerns the Principal Components of the correlation matrix of most popular users and interests. Figure 1 presents the first two Principal Components, while Figure 2 presents Principal Components 3 and 4. Interests are represented by circles, while friends are represented by squares. Likewise, 2004 data are represented by open symbols, while 2005 data are represented by filled symbols. Both figures are enhanced with four ellipses indicating 95% confidence intervals for the distribution of each of the four year/relation categories.

From Figure 1, it is immediately apparent that Principal Components 1 and 2 primarily separate out interests (projecting leftward from the origin) and friends (projecting upward from the origin). Remarkably, there is very little correlation between friends and interests of either year, whereas, across years, friends correlate closely with friends, and interests with interests. The coincidence of the pairs of ellipses for interests and friends further indicates this tendency. Moreover, the locations of interests or friends in a particular year tend to be close to their corresponding locations in the other year. The 2005 locations tend to be a bit closer to the origin overall, indicating weaker correlations among the interests and friends in 2005.

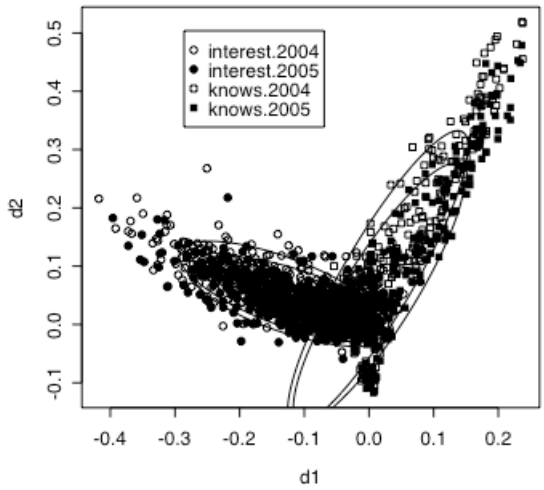

Figure 1. Principal Components 1 and 2 of the interest and friends data of 18,725 LiveJournal users.

Hence, from Figure 1 we learn that the manner in which people elect friends and interests in their LiveJournal profiles is sharply different. This is all the more surprising in that, although popular interests tend to be more common than popular friends, the two sets of relations nonetheless overlap substantially in their frequencies. Conse-

quently we must conclude that these differences are not merely artifacts of the method, but represent fundamentally different social behaviors.

Principal components 3 and 4 confirm and expand on this finding, as seen in Figure 2. Here, there are also two main branches of data points, and again 2005 data points tend to be a bit closer to the origin from their corresponding locations in 2004. This time, the two branches consist entirely of distinct intra-correlated sets of friends. There is a tendency for interests to spread in a similar pattern, but much closer to the origin. The confidence ellipses for the four sets of relations show that the 2004 and 2005 data are again largely coincident in their range of variation. However, neither interests nor friends is significantly stretched along either axis to the exclusion of the other. Further principal components show a more-or-less normal distribution about the origin, with interests clustering closer to the origin than friends.

Hence, popular interests exhibit less variation in their overall distribution than popular friends. At the same time, the election of interests and friends by users is not strongly inter-correlated, although strong intra-correlations of certain sets of users, at least, can be found.

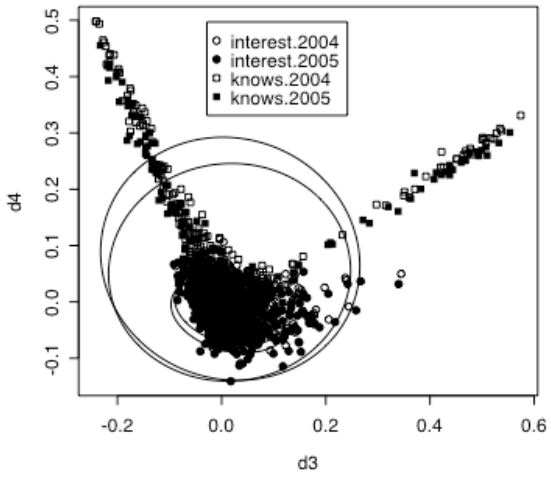

Figure 2. Principal Components 3 and 4 of the interest and friends data of 18,725 LiveJournal users.

Following this, we conducted hierarchical cluster analyses of the four subsets of variables. The clusters found among the interests largely confirm the earlier analysis [5]. At the same time, there is considerable variation from one year to the next in the content of the clusters. For example, Figure 3 shows dendrograms for two roughly corresponding clusters from the foaf:knows relations in 2004 and 2005. The 2004 cluster is much tighter than the 2005 cluster, as indicated by the height scale along the top, while the 2005 cluster is considerably larger, containing more members. Popular users in the 2004 cluster are readily located in the 2005 cluster, although their relative proximities are generally not maintained, and they are intermingled with popular users

not present in the 2004 cluster. Apparently, this cluster became somewhat looser from 2004 to 2005, and expanded into a region of space where other clusters of popular users were located.

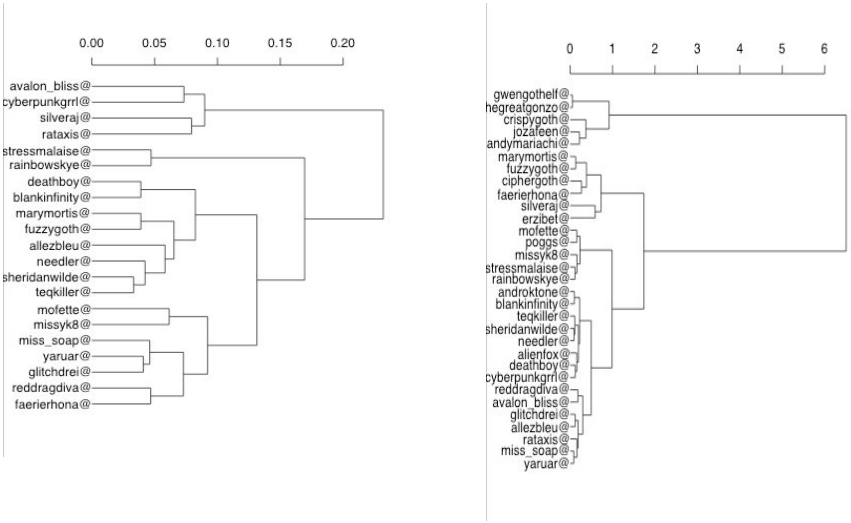

Figure 3. Dendrograms for corresponding clusters of popular users (foaf:knows) in 2004 and 2005.

Because of this movement at the finer levels of structure in the dendrograms, we felt that the analysis would benefit from examining the structures present at a coarser level of granularity. This was done by using a "cut" at a height that would partition the interests and popular users into a small number of clusters. The interests clusters were then interpreted; the knows clusters were not interpreted, since that would have required reading hundreds of weblogs to form a suitable interpretation. Using a cut with five clusters, the 2004 interests can be partitioned as follows:

(1) Science Fiction, Fantasy, Celtic, and graphic arts,
(2) General interests
(3) Sex, goth subculture, body modification and fetish
(4) A variety of contemporary music interests (The Cure, Joy Division, David Bowie, Radiohead, Pink Floyd, Led Zeppelin, Ani DiFranco, Modest Mouse, etc.)
(5) Industrial and alternative rock music.

The 2005 interests are somewhat different, partitioning into clusters as follows:

(1) Social, natural, and mystical interests
(2) General interests
(3) Ska, punk and alternative rock music
(4) Science Fiction, Fantasy, Celtic, and graphic arts
(5) Sex, goth subculture, body modification and fetish

There is clearly a broad correspondence among the categories of both years, with some of the clusters (such as Cluster 1 in 2004 and Cluster 4 in 2005) nearly identical across the two years. At the same time, some differences are evident. To ascertain the extent and nature of the reorganization of interest clusters, we cross-tabulated the analyses for the two years and re-arranged rows and columns to maximize the diagonal. Thus, we equate clusters from the two years that have maximal common membership. We then visualized this as a network diagram, using line weight to represent the strength of a link. Self-links (loops) indicate the size of the membership retained from 2004 to 2005. This network is presented in Figure 4.

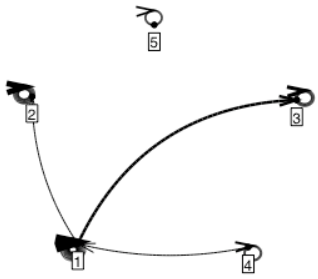

Figure 4. Exchange of members of five interest clusters from 2004 to 2005.

It is evident that most of the clusters have a fairly constant membership, although there is some movement from the largest cluster, general interests (1) into the second and third largest clusters. At the same time, there is some movement of membership from the fourth group into the first. This is because Cluster 1 is something of a grab-bag cluster, whose members lie reasonably close to the origin of the Principal Components of the interests. Hence, we see what might be a strengthening of at least certain interest clusters from 2004 to 2005, whose membership increases at the expense of the largest category of relatively undifferentiated interests.

In earlier work, we also relied on partitions into much larger numbers of interest clusters to develop interpretations [5]. However, in comparing the 2004 and 2005 cluster analyses, we noted a great deal of movement within most of the five clusters. For example, in the cluster containing sexual interests, we observed that most of the sub-clusters of sexual interests had completely re-organized between the 2004 and 2005 analyses. We consider it unlikely that the social or semantic categories of sexual (or other) interests has truly changed to this extent over the course of a single year, for this set of LiveJournal users. Rather, we propose that a user's nomination (or modification) of interests is subject to some random variation, which we observe in these intra-category shifts. Data from additional years would likely help in ascertaining the extent of this variation.

We conducted a similar analysis of the foaf:knows clusters, taking a cut that yielded six clusters of popular users. It is much harder to characterize the commonality among a cluster of users — what they represent is a group of weblogs that are visited and read by a similar set of users, so they cannot be interpreted as readily as lists of interests. However, it does appear that at least some of the clusters have a social coherence, for example, one cluster appears to be a network of goth/erotic models and photographers based in Canada and Virginia, and another appears to be a group of Spanish-speaking photographers.

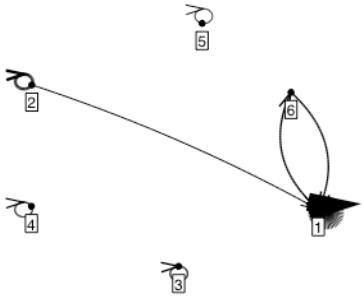

Figure 5. Exchange of members of six friends clusters from 2004 to 2005.

The 2004 and 2005 friends clusters were submitted to a network analysis in the manner of the preceding analysis for interests; this is represented in Figure 5. What is notable about Figure 4 is its relative stability. There is some accretion of popular users from the second-largest group into the largest group, but beyond this, all the groups but one largely maintain their membership. The only exception is Cluster 6, which has an apparent 100% turnover in membership. This is readily explained as an unstable cluster identification that is not distinct from Cluster 1. Had the cut been set at a higher threshold, only five clusters would have been found, with the membership of Cluster 6 for both years included in Cluster 1.

In addition to these two analyses, we made a number of attempts to illustrate the relations among the clusters of interests and clusters of friends, none of which produced a revealing set of patterns, regardless of the similarity measure used (e.g. Euclidean distance or cosine correlation). For example, a bi-modal network containing the six interest and friends clusters as the two node classes simply shows every interest cluster connected to every friends cluster at approximately the same level of strength. This is to be expected from the lack of correlation of interests and friends found in Figure 1, and in the greater concentration of interests around the origin.

4 Discussion

Among the clusters of popular users, there has been minimal change from 2004 to 2005. Among interests there is somewhat greater change, although still substantial stability, and the differentiation of the interest clusters appears to be strengthening somewhat. These results suggest that, with some caveats, it should be possible to identify clusters of users and interests that could lead to useful inferences for Semantic Web applications. For example, it is not hard to imagine a journal recommender system, or an interest or community weblog recommender system, based on the principles of analysis laid out in this research. This is in fact one way that recommender systems like those of Amazon.com operate.

At the same time, there is a great deal of re-organization of relations among interests at finer levels of distinction, suggesting that there is a limit to the efficacy of inferences we can draw from the clusters. For example, regardless of the strength of correlation between two interests, it would be incorrect to conclude that there is a necessary relation between them, or that in subsequent years the same relationship would obtain. This is important in two ways. First, in semantic domains where ontologies are not available or are unreliable, such as in the categorization of popular music, it will be necessary to supplement or replace logical modes of inference with something else. Statistical analyses of large, socially relevant user preferences is a promising source for additional information. Second, it is probable that other system developers will make use of clustering or related methods for "unsupervised learning" of Semantic Web ontologies or inference rules in these domains.

In either scenario, we depend upon statistical means of inference, which are probabilistic and subject to variation. Hence it is critical that we correctly estimate the expected variation in category assignments arising from different profile data sets. This requires careful consideration; while the literature on evaluating cluster analyses can be a helpful guide here, we also need to understand better the nature of the choice that goes into the editing of user profiles. These considerations are different for friends and interests, and lead to strikingly different distributions, even though they are represented identically in the user interface, in the metadata markup and in the incidence matrix used in our analysis.

Finally, since social and semantic relations do not correlate, it is not obvious that the semantics of interest clusters is emergent from the social relations experienced on LiveJournal. This may be because the 500 most popular friends are something like the local equivalent to celebrities. Users friending such LiveJournal celebrities need not expect to have a direct social relationship with them. Rather, their products (journal entries), like the cultural products of real-life celebrities, are passively consumed. On the other hand, this would suggest that popular users should pattern more like interests. The fact that they do not means that there is something different about these sorts of celebrities and the music celebrities that form the basis of many interest clusters. Hence the social dimensions of these user behaviors, and the semantics associated with them, need further study.

5 Conclusions

This study illustrates several ways in which the application of SNA methods to Semantic Web metadata holds much promise. First, we can use SNA methods to reveal highly structured relations among markup elements that are not themselves part of a controlled vocabulary or RDF vocabulary. Hence, we can use SNA and its statistical methods to extend the inference mechanisms already envisioned for the Semantic Web. In addition, by making diachronic observations, we can examine the extent to which naturally inter-correlated groups of interests or friends are stable over time. In this analysis, we discover distinct patterns of stability and flux for these two types of relation. Hence we recognize a need for caution in basing inferences on the clusters we discover this way, noting the need to develop a more complete understanding of the processes of nominating friends and interests.

References

[1] Passin, T (2004) *Explorers Guide to the Semantic Web*. Manning, Greenwich CT.

[2] Davies, J; Fensel, D; and van Harmelen, F, eds. (2003) *Towards the Semantic Web: Ontology-Driven Knowledge Management*. Wiley, New York.

[3] Berners-Lee T, Hendler J, Lassila O (2001) The Semantic Web. *Scientific American* 284, May: 34-43.

[4] Ding, L; Zhou, L; and Finin, T (2005) FOAF Discovery: A Structural Perspective of the Semantic Web, *Proceedings of the 38th Hawaii International Conference on System Sciences*. IEEE Computer Society, Los Alamitos, California.

[5] Paolillo, J, and Wright, E (2005) Social network analysis on the Semantic Web: Techniques and Challenges for Visualizing FOAF. In A. Geroimenko and C. Chen, eds., *Visualizing the Semantic Web, Second Edition*. Springer, Berlin.

[6] Scott, J (2000) *Social Network Analysis: A Handbook, Second Edition*. Sage Publications, Thousand Oaks, California.

[7] Wasserman S, Faust K (1994) *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge.

[8] Eckert, P (2000) *Linguistic Variation as Social Practice*. Blackwell, Oxford.

[9] Doreian, P; and Fujimoto, K (2002) Structures of Supreme Court Voting. *Connections 25(3)*.

[10] Herring S, Scheidt L, Bonus S, Wright E. (2004) Bridging the Gap: A Genre Analysis of Weblogs. In *Proceedings of the Thirty-seventh Hawaii International Conference on System Sciences* (HICSS-37). IEEE Computer Society, Los Alamitos, California.

[11] Brickley D, Miller L (2003) FOAF Vocabulary Specification. Technical report, RDFWeb FOAF Project.

[12] Landauer, T; Foltz, P; and Laham, D (1998) Introduction to Latent Semantic Analysis. *Discourse Processes,* 25: 259-284.

[13] Bayer, A; Smart, J; McLaughlin G (1990) Mapping intellectual structure of a scientific subfield through author cocitations. *Journal of the American Society for Information Science*, 41(6): 444-452.

[14] White, H (1986) Cocited author retrieval. Information Technology and Libraries, 5(2):93-99.

[15] White, H; and Griffith, B (1981) Core journal networks and cocitation maps in the marine sciences: Tools for information management in interdisciplinary research. *Journal of the American Society for Information Science*, 32(3):163-171.
[16] Basilevsky, A (1994) *Statistical Factor Analysis and Related Methods: Theory and Application*. Wiley, New York.

Analyzing Semantic Interoperability in Bioinformatic Database Networks*

Philippe Cudré-Mauroux, Julien Gaugaz, Adriana Budura, and Karl Aberer

School of Computer and Communication Sciences
EPFL – Switzerland

Abstract. We consider the problem of analytically evaluating semantic interoperability in large-scale networks of schemas interconnected through pairwise schema mappings. Our heuristics are based on a graph-theoretic framework capturing important statistical properties of the graphs. We validate our heuristics on a real collection of interconnected bioinformatic databases registered with the Sequence Retrieval System (SRS). Furthermore, we derive and provide experimental evaluations of query propagation on weighted semantic networks, where weights model the quality of the various schema mappings in the network.

1 Introduction

Even if Semantic Web technologies have recently gained momentum, their deployment on the wide-scale Internet is still in its infancy. Only a very small portion of websites have so far been enriched with machine-processable data encoded in RDF or OWL. Thus, the difficulty to analyze semantic networks due to the very lack of realistic data one can gather about them. In [5], we introduced a graph-theoretic model to analyze interoperability of semantic networks and tested our heuristics on large-scale, random topologies. In this paper, we extend these heuristics and test them on a real semantic network, namely on a collection of schemas registered with the Sequence Retrieval System (SRS).

We start below by giving a short introduction to SRS. We then present our approach, which boils down to an analysis of the component sizes in a graph of schemas interconnected through schema mappings. We report on the statistical properties of the SRS network we consider and on the performance of our heuristics applied on this network. Finally, we report on the performance of our approach on larger and weighted networks mimicking the statistical properties of the SRS network.

* The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

2 The Sequence Retrieval System (SRS)

SRS, short for "Sequence Retrieval System", is a commercial information indexing and retrieval system designed for bioinformatic libraries such as the EMBL nucleotide sequence databank, the SwissProt protein sequence databank or the Prosite library of protein subsequence consensus patterns. It is a distributed, interoperable data management system which was initially developed at the European Molecular Biology Laboratory in Heidelberg, and which allows the querying of one or several databases simultaneously, regardless of their format or schemas.

Administrators wishing to register new databases with SRS first have to define the schema they have adopted to store data, using a custom language called *Icarus*. Once their schemas have been defined, administrators can import schema instances (i.e., text files) whose data will be correctly parsed, indexed and processed thanks to the corresponding schema definitions. Additionally, administrators can manually define relationships between their database schema and similar schemas. In SRS, these relationships are represented as links relating one entry of a database schema to one entry of another schema. Thanks to this structure of links between databases, users can propagate queries they pose locally against one specific schema to other schemas available in the system (for technical details, we refer the interested reader to *http://www.lionbioscience.com/*)

2.1 Graph analysis of an SRS repository

Conceptually, the model described above is very close to what one could expect from a subgraph of the semantic web itself, i.e., a collection of related schemas (or ontologies) linked one to another through pairwise mappings. The graph which can be extracted from a SRS repository has two main advantages over those which can be built from current RDFS / OWL repositories: i) it is based on a real-world collection of schemas which are being used on a daily basis by numerous independent parties and ii) it is of a reasonable size (several hundreds of semantically related complex schemas). Thus, after having been rather unsuccessful at finding reasonable semantic networks from the Semantic Web itself, we decided to build a specialized crawler to analyze the semantic graph of an SRS repository and to test our heuristics on the resulting network.

We chose to analyze the semantic network from the European Bioinformatics Institute SRS repository, publicly available at *http://srs.ebi.ac.uk/*. We built a custom crawler which traverses the entire network of databases and extracts schema mapping links stored in the schema definition files. The discussion below is based on the state of the SRS repository as of May 2005.

The graph resulting from our crawling process is an undirected graph of 388 nodes (database schemas) and 518 edges (pairwise schema mappings). We chose to represent links as undirected edges since they are used in both directions by SRS (they basically represent cross-references between two entries of two database schemas). We identified all connected components in the graph (two nodes are in the same connected component if there is a path from one to

the other following edges). The analysis revealed one giant connected component (i.e., a relatively large set of interconnected schemas) of 187 nodes, which represent roughly half of the nodes and 498 edges. Besides the giant connected component, the graph also has two smaller components, each consisting of two vertices. The rest of the nodes are isolated, representing mostly result databases or databases for which no link to other databases was defined.

The average degree of the nodes is 2.2 for the whole graph and 4.6 for the giant component. Real networks differ from random graphs in that often their degree distribution follows a power law, or has a power law tail, while random graphs have a Poisson distribution of degrees [2]. Unsurprisingly, our semantic network is no exception as can be seen in Figure 1 below, which depicts an accurate approximation of the degree distribution of our network by a power-law distribution $y(x) = \alpha x^{-\gamma}$ with $\alpha = 0.21$ and $\gamma = 1.51$.

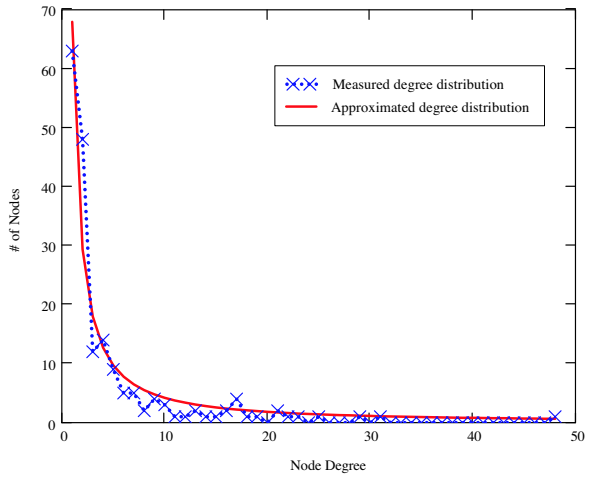

Figure 1: An approximation of the degree distribution of our semantic network by a power-law distribution $y(x) = \alpha x^{-\gamma}$ with $\alpha = 0.21$ and $\gamma = 1.51$

Another interesting property which we explored was the tendency of the schemas to form clusters, quantified by the average clustering coefficient. Intuitively, the clustering coefficient of a vertex measures the degree to which its neighbors are neighbors of each other. More precisely, the clustering coefficient indicates the ratio of existing edges connecting a node's neighbors to each other to the maximum possible number of such edges. The network we considered has a high average clustering coefficient of 0.32 for the whole graph and of 0.54 for the giant component. The diameter (maximum shortest paths between any two vertices) of the giant connected component is 9. These data indicate that our network can be characterized as *scale-free* (power-law distribution of degrees) or *small-world* (small diameter, high clustering coefficient).

3 Analyzing semantic interoperability in the large

In [5], we introduced a graph-theoretic framework for analyzing semantic interoperability in large-scale networks. As described above, we model database schemas (or ontologies) as nodes, interconnected by edges (schema mappings). Schema mappings are used to iteratively propagate queries posed against a local schema to other related schemas (see [1] for how this can be implemented in practice). Note that links can be directed or undirected, weighted or non-weighted depending on the schema mappings being used.

In such a network, the density of mappings is important in order to propagate a local query from one database (schema) to the other databases. A query can only be propagated to all databases if the semantic network is *connected*, that is if there exists a path from one schema to any other schema following schema mapping links. If some schemas are isolated, queries cannot be propagated to/from the rest of the graph, thus making it impossible to have a semantically interoperable network of databases. This observation motivated us to take advantage of percolation theory to determine when a semantic network could be connected or not. Our framework for analyzing semantic interoperability takes advantage of *generatingfunctionologic* [7] functions for the degree distribution of the semantic graph:

$$G_0(x) = \sum_{k=0}^{\infty} p_k x^k \tag{1}$$

where p_k represents the probability that a randomly chosen vertex has degree k. We showed (by extending results from [6]) that a network cannot be semantically interoperable in the large unless the *connectivity indicator* $ci = \sum_k k(k-2-cc)p_k$ is greater than zero, with cc representing the clustering coefficient. Also, we provided heuristics for estimating the relative size S of the biggest semantically interoperable cluster of schemas by solving

$$S = 1 - G_0(u), \tag{2}$$

where u is the smallest non-negative real solution of

$$u = G_1(u) \tag{3}$$

and $G_1(u)$, the distribution of outgoing edges from first to second-order neighbors, is

$$G_1(x) = \frac{1}{x^{cc}} \frac{G_0'(x)}{G_0'(1)} = \frac{1}{z_1} \frac{1}{x^{cc}} G_0'(x). \tag{4}$$

3.1 Applying our heuristics to the SRS graph

We applied our heuristics to the SRS graph we obtained from the crawling process. The results are as follows: we get a connectivity indicator ci of 25.4,

indicating that the semantic network is clearly in a super-critical state and that a giant component interlinking most of the databases has appeared. The size of this giant component as estimated by our heuristics (see above) is of 0.47, meaning that 47% of the schemas are part of the giant connected component. This is surprisingly close to the real value of 0.48 as observed in the graph.

3.2 Generating a Graph with a given Power-Law Degree Distribution

Going slightly further, we want to analyze the dynamics of semantic graphs with varying numbers of edges. Our aim is to generate graphs with the same statistical properties as the SRS graphs, that is, graphs following a power-law degree distribution:

$$P(k) = \alpha k^{-\gamma} \tag{5}$$

but with a varying number of edges. We take from [3] a graph-building algorithm yielding a power-law degree distribution with a given exponent γ. It goes as following: (1) create a (large) number N of vertices. (2) to each vertex i, assign an "importance" x_i, which is a real number taken from a distribution $\rho(x)$. (3) for each pair of vertices, draw a link with probability $f(x_i, x_j)$, which is function of the importance of both vertices.

Now if $f(x_i, x_j) = (x_i x_j / x_M^2)$ (where x_M is the largest value of x in the graph), we know from [3] that the degree distribution of a graph will be

$$P(k) = \frac{x_M^2}{N\langle x\rangle}\rho\left(\frac{x_M^2}{N\langle x\rangle}k\right) \tag{6}$$

where $\langle x\rangle$ is the expected value of the importance x, such that $P(k)$ follows a power-law if $\rho(x)$ does so.

We then choose a power-law distribution

$$\rho(x) = \frac{\gamma - 1}{(m^{1-\gamma} - Q^{1-\gamma})}x^{-\gamma} \tag{7}$$

defined over the interval $[m, Q]$. However, we still have to find values for m and Q such that the scale of the resulting degree distribution equals α. Using equation 7, we find the expected importance value as

$$\langle x\rangle = \frac{(\gamma - 1)(m^2 Q^\gamma - m^\gamma Q^2)}{(\gamma - 2)(mQ^\gamma - m^\gamma Q)}. \tag{8}$$

Replacing $\rho(x)$ in equation 6, the degree distribution of the resulting graph becomes

$$P(k) = \frac{x_M^2}{N\langle x\rangle}\frac{\gamma - 1}{(m^{1-\gamma} - Q^{1-\gamma})}\left(\frac{x_M^2}{N\langle x\rangle}k\right)^{-\gamma} \tag{9}$$

such that, equating with equation 5, we get

$$\alpha = \frac{x_M^2}{N\langle x\rangle} \frac{\gamma - 1}{(m^{1-\gamma} - Q^{1-\gamma})} \left(\frac{x_M^2}{N\langle x\rangle} \right)^{-\gamma}. \quad (10)$$

We can then arbitrarily choose $m > 0$ and find Q by numerical approximation, since the right-hand side of equation 10 is defined and continue for values of $Q > m$.

Figures 2 and 3 show the results of our heuristics on networks of respectively 388 (i.e., mimicking the original SRS graph) and 3880 edges (i.e., 10 times bigger than the original SRS graph but with the same statistical properties) constructed in the way presented above with a varying number of edges. The curves are averaged over 50 consecutive runs. As for the original SRS network, we see that we can accurately predict the size of the giant semantic component, even for very dense graphs.

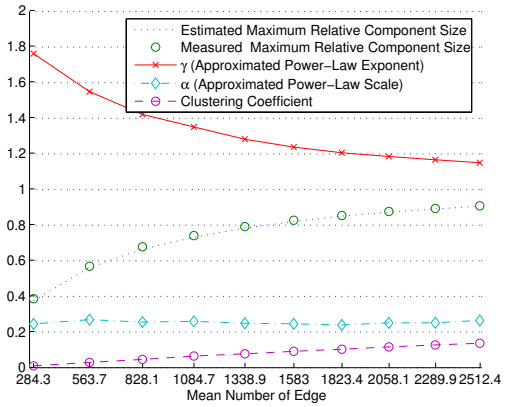

Figure 2: Estimating the giant component size of a scale-free semantic network of 388 nodes with a varying number of edges

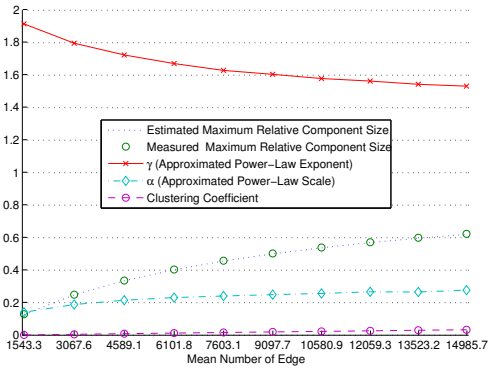

Figure 3: Estimating the giant component size of a scale-free semantic network of 3880 nodes with a varying number of edges

4 Connectivity Indicator and Giant Component Size in Weighted Graphs

So far, we only analyzed the presence and size of a giant connected component in order to determine which portion of a semantic network could potentially be semantically interoperable. In large-scale decentralized networks, however, one should not only look into the giant semantic component itself, but also analyze the *quality* of the mappings used to propagate queries from one schema to the other (see [1] for a discussion on that topic). Indeed, in any large, decentralized network, it is very unlikely that all schema mappings could *correctly* map queries from one schema to the other, because of the lack of expressivity of the mapping languages, and of the fact that some (most?) of the mappings might be generated automatically.

Thus, as considered by more and more semantic query routing algorithms, we introduce weights for the schema mappings to capture the quality of a given mapping. Weights range from zero (indicating a really poor mapping unable to semantically keep any information while translating the query) to one (for perfect mappings, keeping the semantics of the query intact from one schema to the other). We then iteratively forward a query posed against a specific schema to other schemas through schema mappings if and only if a given mapping has a weight (i.e., quality) greater than a predefined threshold τ. $\tau = 0$ corresponds to sending the query through any schema mapping, irrespective of its quality. On the contrary, when we set τ to one, the query gets only propagated to semantically perfect mappings, while even slightly faulty mappings are ignored. Previous works in statistical physics and graph theory have looked into percolation for weighted graphs. We present hereafter an extension of our heuristics for weighted semantic networks inspired by [4].

4.1 Connectivity Indicator

As before, we consider a generating function for the degree distribution

$$G_0(x) = \sum_{k=0}^{\infty} p_k x^k \tag{11}$$

where p_k is the probability that a randomly chosen vertex has degree k in the network. We then define t_{jk} as the probability that an edge has a weight above τ given that it binds vertices of degree j and k. Thus, $w_k = \sum_{j=0}^{\infty} t_{jk}$ is the probability that an edge transmits, given that it is attached to a vertex of degree k. The generating function for the probability that a vertex we arrive at by following a randomly chosen edge is of degree k *and* transmits is

$$G_1(x) = \frac{\sum_{k=0}^{\infty} w_k k x^{k-1}}{x^{cc} \sum_{k=0}^{\infty} k p_k} \tag{12}$$

where *cc* is the clustering coefficient. Now, from [4], we know that the generating function for the probability that one end of a randomly chosen *edge* on the graph

leads to a percolation cluster of a given number of vertices is

$$H_1(x) = 1 - G_1(1) + xG_1\left[H_1(x)\right]. \tag{13}$$

Similarly, the generating function for the probability that a randomly chosen *vertex* belongs to a percolation cluster of a given number of vertices is

$$H_0(x) = 1 - G_0 + xG_0\left[H_1(x)\right] \tag{14}$$

such that the mean component size corresponding to a randomly chosen vertex is

$$\langle s \rangle = H_0'(1) = G_0(1) + \frac{G_0'(1)G_1(1)}{1 - G_1'(1)} \tag{15}$$

which diverges for $G_1'(1) \geq 1$. However,

$$G_1'(1) = \frac{\sum_{k=0}^{\infty} w_k p_k k(k - 1 - cc)}{\sum_{k=0}^{\infty} k p_k} \tag{16}$$

such that a giant connected component appears if

$$ci = \sum_{k=0}^{\infty} k p_k \left[w_k(k - 1 - cc) - 1\right] \geq 0 \tag{17}$$

4.2 Giant Component Size

As seen above, $H_0(x)$ represents the distribution for the cluster size which a randomly chosen vertex belongs to, *excluding* the giant component. Thus, according to [4], $H_0(1)$ is equal to the fraction of the nodes which are not in the giant component. The fraction of the nodes which are in the giant component is hence $S = 1 - H_0(1)$. Using equation 14 we can write

$$S = 1 - H_0(1) = G_0(1) - G_0\left[H_1(1)\right]. \tag{18}$$

with $H_1(1) = 1 - G_1(1) + xG_1\left[H_1(1)\right]$. Thus $H_1(1) = u$ where u is the smallest non-negative solution of

$$u = 1 - G_1(1) + G_1(u). \tag{19}$$

The relative size of the giant component reached by the query in a weighted semantic graph follows as

$$S = G_0(1) - G_0(u). \tag{20}$$

Figures 4 and 5 show the results of our heuristics on weighted networks of respectively 388 and 3880 nodes, for a varying number of edges and various values of τ. The curves are averaged over 50 consecutive runs, and the weights of

individual schema mappings are randomly generated using a uniform distribution ranging from zero to one. We see that our heuristics can quite adequately predict the relative size of the graph to which a given query will be forwarded. Also, as for the unweighted analysis, we observe similar behaviors for the two graphs; This is rather unsurprising as we are dealing with scale-free networks whose properties are basically independent of their size.

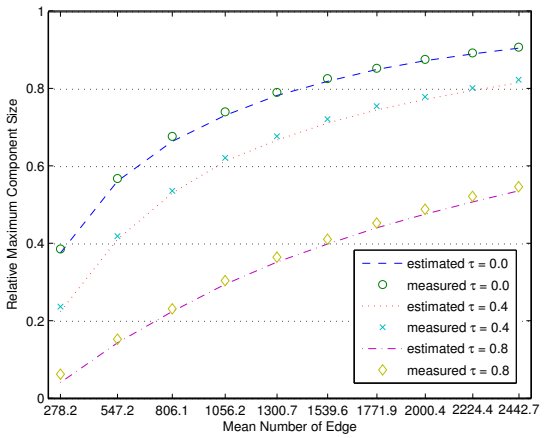

Figure 4: Fraction of the graph a local query will be forwarded to, for a weighted network of 388 nodes with a varying number of edges and various forwarding thresholds τ

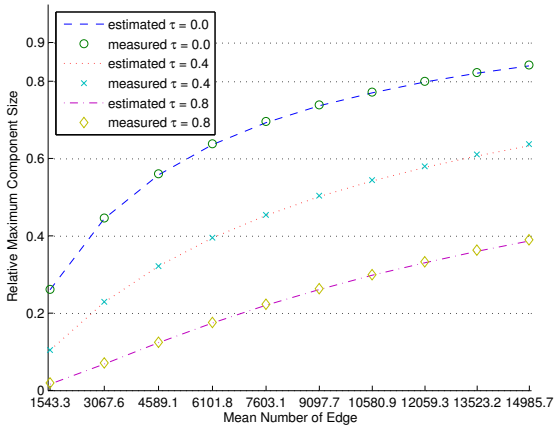

Figure 5: Fraction of the graph a local query will be forwarded to, for a weighted network of 3880 nodes with a varying number of edges and various forwarding thresholds τ

5 Conclusions

In this paper, we tested graph-theoretic heuristics to evaluate semantic interoperability on a real semantic network. The results confirm the validity of our heuristics beyond our initial hopes as we could predict quite accurately the size of the giant semantic component in the network. Also, we extended our analysis to apply our heuristics on larger networks enjoying similar statistical properties and on weighted semantic networks. It was for us quite important to test our heuristics using real-world data, as semantic network analyses mostly consider artificial networks today, due to the current lack of semantically enriched websites or deployed semantic infrastructures. In the future, we plan to extend our analyses to take into account the dynamicity (churn) of the network of schema mappings, and to consider more accurate query forwarding schemes based on transitive closures of mapping operations.

References

1. K. Aberer, P. Cudré-Mauroux, and M. Hauswirth. The Chatty Web: Emergent Semantics Through Gossiping. In *International World Wide Web Conference (WWW)*, 2003.
2. R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(47), 2002.
3. G. Caldarelli, A. Capocci, P. D. L. Rios, and M. Muoz. Scale-free networks from varying vertex intrinsic fitness. *Phys. Rev. Lett.*, 89, 258702, 2002.
4. D. S. Callaway, M. Newmann, S. H. Strogatz, and D. J. Watts. Network robustness and fragility: Percolation on random graphs. *Phys. Rev. Lett.*, 85, 54685471, 2000.
5. P. Cudré-Mauroux and K. Aberer. A Necessary Condition For Semantic Interoperability In The Large. In *International Conference Ontologies, DataBases, and Applications of Semantics (ODBASE)*, 2004.
6. M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Phys. Rev.*, E64(026118), 2001.
7. H. S. Wilf. *Generatingfunctionology.* 2nd Edition, Academic Press, London, 1994.

Ontologies are us: A unified model of social networks and semantics

Peter Mika

Vrije Universiteit, Amsterdam
1081HV Amsterdam, The Netherlands
pmika@cs.vu.nl

Abstract. On the Semantic Web ontologies are most commonly treated as artifacts created by knowledge engineers for a particular community. The task of the engineers is to forge a common understanding within the community and to formalize the agreements, prerequisites of reusing domain knowledge in information systems. However, the process of objectifying ontologies results in ontological drift over time (as the community and its understanding evolves independently of the agreement) and results in the loss of local views over the domain.

Several authors have suggested *emergent semantics* as a solution [1]. The idea of emergent semantics is to define the ontology as an emergent feature of a system of autonomous agents acting in dynamic, open environments. Besides an agreement over the kind of environment in which emergence could be observed, there is little common ground in what emergence would constitute.

We have proposed elsewhere that the first step towards emergent semantics is a representation of ontologies that incorporates the social context of concepts and their use [2]. In this work, we propose an abstract model that extends the traditional conceptualization of ontologies with the social dimension, leading to a tripartite model of actors, concepts and instances. We show how simple graph transformations can be applied to such a semantic social network for obtaining two different kinds of ontologies (semantic networks): one ontology based on overlap in the sets of instances of concepts and another based on the overlap of communities who have applied those concepts.

We demonstrate the significant differences between these networks by applying our model in two separate case studies. First, we investigate a large scale semantic social network, the del.icio.us social bookmarking tool. We analyze the network properties of the two ontologies and show how clusters of related concepts and taxonomical relationships can be extracted to enrich the representation. Second, we apply our ideas toward extracting community ontologies from the Web, i.e. semantic networks that reflect the understanding of a particular community on the Web. We evaluate this method against the results of traditional web mining using co-occurrence analysis. The results show that the emergent semantic network is more accepted by the members of the community, especially those closer to the core of the community.

References

1. Karl Aberer, Philippe Cudré-Mauroux, Aris M. Ouksel, Tiziana Catarci, Mohand-Said Hacid, Arantza Illarramendi, Vipul Kashyap, Massimo Mecella, Eduardo Mena, Erich J. Neuhold, Olga De Troyer, Thomas Risse, Monica Scannapieco, Fèlix Saltor, Luca de Santis, Stefano Spaccapietra, Steffen Staab, and Rudi Studer. Emergent Semantics Principles and Issues. In *Database Systems for Advanced Applications 9th International Conference, DASFAA 2004*, volume 2973 of *LNCS*, pages 25–38, 2004.
2. Peter Mika. Social Networks and the Semantic Web: The Next Challenge. *IEEE Intelligent Systems*, 20(1), January/February 2005.

From the Semantic Web to Web 2.0 - Semantic Web for Social Networks

Stefan Decker

Digital Enterprise Research Institute
National University of Ireland
Galway, Ireland
stefan@stefandecker.org

Abstract. More recently a new research stream is getting more and more attention: the application of Semantic Web technology for social collaborative software like Blogs or Wikis. In this talk I will summarize recent developments and provide technology examples for the application of Semantic Web for social networks. Examples will especially include Semantic Blogging, Semantic Wikis and the Social Semantic Desktop. Additionally I will provide my view how Social Network Analysis can benefit from the current trend and provide valuable data users exploiting these upcoming technologies. Examples will include recommender systems exploiting social relationships, but my talk will also go into more speculative directions.