Corporate Smart Content

Designs and Prototypes

Report IV on Knowledge-based Mining of Complex Event Patterns:
Semantic Pattern Mining in Event Streams

Technical Report TR-B-16-01

Ahmad Hasan, Yamen Jeries, Ahmed Nader, Adrian Paschke

Freie Universität Berlin
Department of Mathematics and Computer Science
Corporate Smart Content

March 15, 2016

GEFÖRDERT VOM

Abstract

In this report, we present our research results for the fourth half-year phase of the project Corporate Smart Content under the working package "Knowledge-based Mining of Complex Event Patterns".

We present SpaceROAM, our new approach to Role Discovery that focuses on the domain of Collaborative Content Management Systems. The method tries to magnify users' characteristic features by positioning them among other one of the same user as well as among features of other users. Applied on real-life data, the method was able to recognize all typical roles and to detect new ones that differ from the known ones significantly.

Finally, we provide a description of other research tasks that we started and whose results will be included in our next report.

1 Role Discovery in Collaborative Content Management Systems

Collaborative Content Management Systems enable users to join their efforts and build content of interest collaboratively. The content itself provides the medium through which members of the community interact [13]. The freedom provided by today's collaborative systems makes organization of the process of content construction difficult since users' domains, tasks and rights are usually not controlled.

On the other hand, role discovery allows grouping users according to the interaction patterns their past activities reflect.

Role Discovery shares considerable similarities with other fields like Topic Modeling and Community Discovery.

The goal of *Topic Modeling* is to categorize a set of documents into different themes[10]. Instead of focusing on users and their interactions, topic modeling studies documents and their content and can be applied to extract non-relational features [14] like the users' topics of interest.

Community Discovery aims at clustering users from a different perspective by looking for groups of strongly connected users. A community is a subnetwork [1] containing entities that are closer to each other than to the entities in the rest of the network [5].

Knowing the roles forming in the user base and identifying users who belong to a specific role can help organizers responsible for collaborative content management, e.g. content managers or team supervisors, coordinate the efforts of the whole community to achieve more efficiency and quality by controlling the contributions and providing more personalized motivations for the users.

For example, active users might be rewarded while less contributing users might be motivated when their behavioral patterns show that they are stuck in some undesirable role. Moreover, it is desirable for a community to have a healthy mix of different roles. Knowing about existing roles in the community and monitoring changes in their distribution would help managers decide how to influence the behavior of their users.

1.1 Related Work

Role Discovery starts from a graph in which nodes are the actors in the system and edges represent interactions among the users. Role discovery is the study of such interaction graphs to find groups of structurally similar nodes.

A comprehensive survey conducted by Rossi and Ahmed in was presented in [14]. Depending on the way nodes are compared and grouped, we distinguish between three main approaches to Role Discovery[14]:

- *Graph-based Roles* are recognized directly from the interaction graph by building an adjacency matrix.
- *Feature-based Approaches* summarize graph-related properties of the nodes to form a feature vector for each node. Nodes are then compared to each other based on their feature vectors and not on their connections within the graph.
- *Hybrid Approaches* mix the previous methods by applying some graph-based algorithm before constructing the feature vector.

In contrast to graph-based approaches that consider node equivalence in a graph [16] [7], feature-based approaches [14] [2] [15] builds a large set of features to be used as the basis for role assignment. Unsupervised approaches to role discovery have been proposed [11] [12], but supervision still has advantages. In [9], Gilpin et al. presented a supervision framework that enforces desired properties like sparsity, diversity and alternativeness.

In our approach, relevant features are calculated for each user and the resulted matrix is normalized twice to give more weight to the user's characteristic features. The supervision is limited to the selection of relevant features.

1.2 Roles in Collaborative CMSs

In a collaborative system with a set U of n users, all user activities are registered in an edit-log that shows changes committed by users on specific pages. An entry in such a log contains at least three attributes: A time stamp, a user ID and a Page identifier, though richer logs exist that show for example the domain of the page or the changed content for each commit operation.

An interaction between two users $u_1, u_2 \in U$ occurs when user u_2 edits an item that was last edited by user u_1. We use this information to build a directed graph $G = \{E, N\}$ in which nodes N represent users or actors in the system and edges E refer to interactions among the users.

Moreover, in feature-based role discovery, we assume a set of features for each user. A feature is usually a structural attribute of the user's node, i.g. the count of incoming or outgoing connections, but may also be a non-structural property [14].

After defining f relevant features and calculating their values for each user, the graph G is no more needed and all further computations can be done on the feature matrix $F \in \mathbb{R}^{n \times f}$.

1.2.1 Typical Features and Roles

Experienced content managers who use Confluence, a collaborative software from Atlassian, in corporate domain suggested some patterns to start our research with. In a collaborative CMS, the following features are usually of interest:

- *Created pages*: The number of pages created by the user. In an edit-log, this case is recognized when a page appears in the log file for the first time.
- *Edited pages*: The number of different pages the user edits.
- *Count of domains*: Assuming the domain of the edited page is provided, users might be evaluated according to the count of different domains they work on.
- *Committing frequency*: Adding a time dimension, the number of changes done by one user per day can be observed to find commitors of the month for example. While the number of edited pages refers to the count of unique pages the user has edited, a commit happens each time the user saves her changes on a page.
- *Cooperations*: The number of pages that return to the user after being edited by other users. Such cases is recognized by the existence of cycles in the interaction graph.
- *Partners*: the number of other users with whom the user cooperates.

The following features require access to the content edited by the users in each commit:

- *Size of changes*: the total number of characters in the body of the commit entry.
- *Referred pages*: The number of other pages the new content refers to.

Moreover, the following patterns are often encountered in collaborative environments:

- *Gardeners* only look around and make little enhancements on the pages of the others. Gardeners feel responsible for the overall quality and appearance of the content. They maintain and update existing information.
- *Readers* who just read pages without changing the content. Detecting this pattern requires data about viewed pages.
- Users who always edit their pages after someone else has edited them. They try to prevent other users from having credit for the final version.

The patterns provided above are of great interest for content managers. However, it is unclear how to systematically uncover hidden patterns in a collaborative system starting, for example, from only one role and some features. Detecting one role, such as the gardeners, is straightforward when relevant features can be identified. Yet the challenge is in discovering new roles that

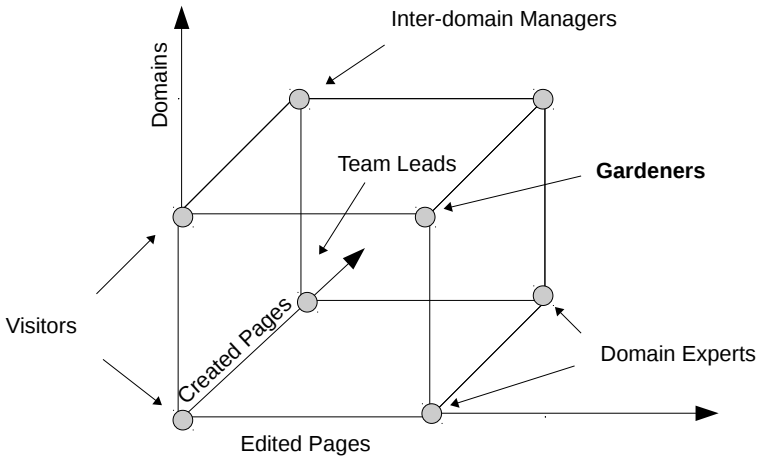


Figure 1: Role Cube: Corners of the three-featured space with their labels.

complement the known ones. In this section, we discuss this problem and present our approach to solving it.

Gardeners, as defined above, make small changes on pages of the others. They tend to edit available content rather than to create pages. The number of domains might also be relevant since edited pages are not necessarily of the same domain. Indeed, gardeners typically take care of formal aspects of the content like format and readability. Hence, we can expect gardeners to have a relatively high number of domains compared to other users.

1.3 SpaceROAM: The Approach

Known algorithms for Role Discovery include the expensive step of clustering the users to find their roles. SpaceROAM avoids this generalization of Role Discovery as a clustering problem and benefits from the observation that meaningful roles must distinguish themselves from other roles by having extreme values for some of the features.

Moreover, clustering users regards their positioning among other users without considering the characteristicness of single features for one user.

SpaceROAM not only considers the feature value for a user among other users, but also magnifies those features that give the users their identity and eventually their roles.

If we go back to the *gardeners* definition in Section 1.2.1, the set of relevant features includes the number of domains and the numbers of edited and created pages. In a vector space generated by the normalized values for those features, gardeners are users residing in a corner with a low number of *created* pages and high numbers of *edited* pages and *domains*.

Each corner of this compact space represents an extreme configuration of the features which is likely to correspond to a role. Figure 1 shows our label suggestions for the other corners in this space.

For instance, users who create pages in a small number of domains without editing their content are team leads who plan the content by bootstrapping empty pages giving them titles and letting other users, namely domain experts, add the real content. Domain experts, on the other hand, can be found at another corner of our space with a high number of edited pages, low number of the created ones and a low number of domains.

Figure 1 depicts a cube made of the corners of the feature space. In addition to the roles described above, each corner in the space corresponds to a potential role with its own characteristic combination of feature values. In this figure, we can find *Team Leads* at the corner with many created pages and low number of domains and edition.

Domain Experts tend to edit pages in a small number of domains adding the real content to pages created by their team leads. *Visitors* are users with little fingerprint in terms of edited and created pages, though they might be active in multiple domains.

To solve the problem of role discovery, SpaceROAM considers a vector space whose dimensions are the relevant features with normalized values. Such a space has f dimensions, each of which having values in the range [0..1].

To consider the user's position among other users without overlooking the distribution of the user's own features, SpaceROAM applies two steps of normalization. In *vertical normalization* shown in equation 1, each feature is scaled for all users to force the range [0..1] resulting in the new feature matrix F'. After this transformation and for each feature, there will be a user with the value one, i.e. the new maximum value after normalization, and a user having the value 0, while other users will be distributed over the range [0..1].

Now we can focus on the feature strength for a single user. For instance, a user with a low number of domains might still be a gardener if the values of other features are even lower because this makes the domains, though being low compared to other users, characteristic for this user.

To take this into account, we apply a *horizontal normalization* shown in equation 2 that forces the range [0..1] on the features of each user separately giving her strongest feature a value of one and the weakest a value of zero. We call the resulting matrix F''. This normalization step gives greater weight to characteristic features and pushes the user to some border of the feature space.

The final step is to assign the roles to users based on their position in the feature space. We simply round all features for each user to zero or one and give numbers according to the resulting binary combination as shown in equation 3. Users laying at some edge of the space will now be pushed to a corner at which each feature has a value of either 0 or 1.

$$F'_{ij} = \frac{F_{ij} - fMin_j}{fMax_j - fMin_j} \tag{1}$$

$$F''_{ij} = \frac{F'_{ij} - uMin_i}{uMax_i - uMin_i} \tag{2}$$

$$R_i = \sum_{j=1}^{f} 2^{j-1} \times \lceil F''_{ij} \rceil \tag{3}$$

Where:

- F' is the feature matrix after the vertical normalization on feature level.
- F'' is the feature matrix after the horizontal normalization on user level.
- i is the user index $\in [1..n]$.
- n is the number of users.
- j is the feature index $\in [1..f]$.
- f is the number of features.
- $fMin_j$ the minimum value of feature j among all users before normalization.
- $fMax_j$ the maximum value of feature j among all users before normalization.
- $uMin_i$ the minimum feature value for user i.
- $uMax_i$ the maximum feature value for user i.
- R_i is the numerical label for the role of user i.
- F''_{ij} the normalized value of j_{th} feature for user i.

Equation 3 simply gives each user a numerical label that corresponds to his corner in the feature space. Users residing at the same position will get the same role label.

1.4 Evaluation

We were provided with an edit-log from a Wiki system that was used to maintain information about available resources, hardware items, minutes of meeting and other internal contents. The dataset contained about **80,000 entries** each referring to a commit operation. The log shows activities of **70 users** who edited a total of **8,734 pages** over a period of **10 years**.

Figure 2 shows the distribution of three of the features we described in Section 1.2.1. For each user, we calculated the number of pages she created or edited as well as the count of distinctive domains to which those pages belong.

Before applying our method on the dataset, we first cleaned the data from noise by excluding users who edited less than 10 pages. Figure 3a shows the distribution of features before the normalization steps. To the right, we see in Figure 3b how applying the two normalization steps pushes the users to the edges of our three-dimensional space magnifying characteristic features of each user.

Applying equation 3 to give numerical labels to the role of each user resulted in the following table:

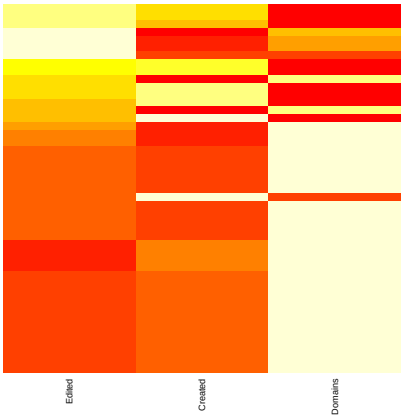
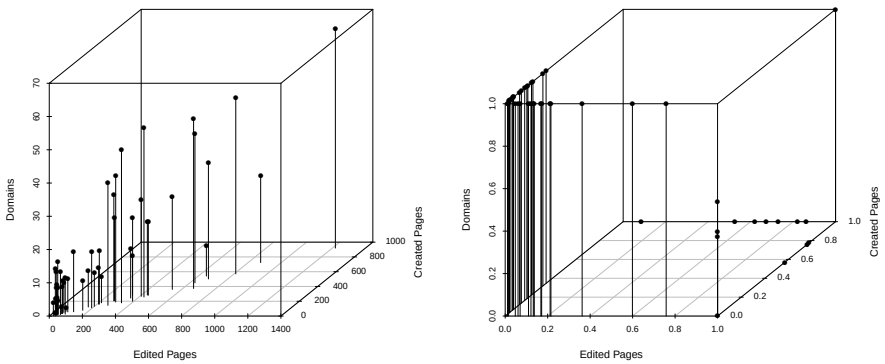


Figure 2: Distribution of selected three features over users



(a) Raw data from edit-log

(b) Data after normalization

Edited	Created	Domains	Role	Users
1	0	0	1	3
0	1	0	2	1
1	1	0	3	9
0	0	1	4	31
1	0	1	5	3
1	1	1	7	1

Table 1: Role labels resulting from applying SpaceROAM on our dataset

The table shows that most of the users were categorized under role 4 which has a high number of domains and few contributions. The next most frequent role is number 3 with many contributions in few domains. Nevertheless, we recognize some users who feature roles we discussed in Section 1.2.1. Indeed, our dataset contains three gardeners in the role labeled 5 (High edits in many domains), three domain experts under role 1 (High edits in few domains) and one team lead under role 2 (Many created pages in few domains).

1.5 Conclusion

SpaceROAM is a simple approach to Role Discovery. It allows exploring existing roles from different perspectives by changing the set of relevant features. This selection step is the only supervision involved in the procedure.

For some set of relevant features, the method can only to roles that reside on the corners of the feature space and does not allow, in the version we presented here, for finer granularities under which roles can also be found in other positions in the space.

In a noisy environment, our method is sensitive to outliers. In the normalization steps, outliers determine the range of the features and affect the normalized values in the resulting normalized matrices for all users and features. This could be solved by applying other types of normalization, i.e. the quantile normalization, and adapting the method accordingly.

In future publications, we plan to set SpaceROAM to test and compare it to other approaches. We will also apply it to larger datasets, like the change logs of the Wikipedia. Moreover, we plan to investigate its potential in stream configuration.

2 Running Work

In this section, we describe two running research tasks we are currently working on. Both works are being accomplished as B.A. theses and are planned to get done before the end of our project in July.

2.1 Distributed Entity Resolution and Disambiguation of Author Names

Digital Libraries and Collections of Information contain nowadays huge amount of documents that is reaching new dimensions. As new technologies appear, solutions for Entity Resolution and Author Name Disambiguation (AND) are gaining more sophistication and complexities. However, none of them had succeeded in creating a complete legitimate engine that combines author names with their unique id's. One of the obstacles that prevented such projects from breaking through, is the perplexity of applying an infrastructure to analyze these huge amounts of published scientific papers from all around the globe.

Distribution Systems, such as Apache Flink with it's streaming and batch processing, offers such a robust and scalable engine to execute these kind of problems with better performance and costs.

2.1.1 Background

Many Papers were and are still being published on Entity Resolution but we will focus in our research on how we can apply these old approaches in a new Distribution System that is yet barely tapped.

Brizan, David Guy, and Abdullah Uz Tansel [17] wrote a survey Entity Resolution and Record Linkage Methodologies and categorized the methodologies into techniques and applications, such as Establishing Match Criteria, TF/IDF, Clustering, Brute Force Applications, Canopy, Bucketing and many other Machine learning algorithms.

Not many semantic researches are known for directly solving the disambiguation of author names, but some efforts were made by Bertin and co. [6] [3] [4] have published some papers mainly on the increased possibility for automated semantic analysis of sentences containing bibliographic references. They propose a method for the exploitation of the full text content of scientific publications through the enrichment of bibliographic metadata harvested by the OAI protocol (The Open Archives Initiative Protocol for Metadata Harvesting). They used a method for automatic annotation and full text semantic analysis specifically designed for scientific publications processing, in order to design tools that offer new functionalities for more efficient exploitation of scientific literature that correspond to specific user needs. Additionally, Hassell, Aleman-Meza and Arpinar [8] worked on Ontology-Driven Automatic Entity Disambiguation in Unstructured Text and proposed, regardless to the structure of the document, a method that uses different relationships in a document as well as from the ontology to provide clues in determining the correct entity. In this way, they could disambiguate names of authors appearing in a collection of DBWorld posts using a large scale, real- world ontology extracted from the DBLP bibliography website. Their precision and recall measurements provided encouraging results.

2.1.2 Data

The raw dataset is originally derived from Song's 8 research. The dataset contained only id's and main author name. For our purposes, we extracted more metadata from the PubMed website using OpenCSV and Jsoup Libraries, and we have now a complete Dataset of 2,875 publications by 385 real authors with 431 name variants. Every publication has the attributes of Author-id, PubMed-id, Main Author Name, Author additional information (such as University, Faculty or Research Center), Co-Authors, Title, Abstract and Journal. Unfortunately, we were not able to get the full texts of these publications from PubMed, as PubMed does not offer such a feature, but we consider this as a chance to exploit our techniques and take advantage of this vulnerability by testing Text Classification Techniques using another dataset that has full texts assigned to their authors.

Every dataset will be divided into two halves in order to have a Training-Set to build and train our modules with the right approaches, and a Test-Set to assess the strength and utility of a predictive relationships.

2.1.3 Work Plan

In this thesis, we will explore the Apache Spark and its API, and according to the literature we will apply an appropriate algorithm from the Flink-ML Library in order to solve our problem. In addition to the Strong Features (e.g co-authors) that we have in our database, we will add features by using Entity Resolution techniques and analyze the texts in order to give every unique name it's own semantic fingerprint, which should help in the process of disambiguating names. We are planning to test multiple solutions in parallel, for example, we have an extra dataset "Reuters-50-50", which is offered Online. Its Training-Set offers 50 Authors and 2,500 texts (50 per author), while its Test-set has 2,500 texts (50 per same author) non-overlapping with the training texts. This dataset offers us a good chance to test our Entity Resolution technique so we can apply it on the abstracts that we have in our first dataset.

2.1.4 Solution Integration and Open Challenges

We might in some cases face some minor challenges, but we will try our best to tackle them and propose a solution that will satisfy our goal. For example, some authors who were never mentioned in our database before, or they could be new and just published their first paper. For that we will try to build an adaptive utility that learns new authors and assign to them new Id's for future classification. Another challenge could be the integration of our module into an enterprise application (such as digital libraries). In such case, we can only imagine a trigger that is fired from the database towards our module in order to get in action.

2.2 Scalable Semantic Enrichment of Event Streams

Semantic Web is a way to let the machines understand the content of web pages by linking data on web pages to entities which can be interpreted by machines. This would allow sharing and combining data from different sources. DBpedia

is the Semantic Web mirror for Wikipedia. It builds a linked data set using automated extraction algorithms. There are many annotation services such as DBpedia Spotlight which is a tool for annotating unstructured natural language by providing references to the linked data on DBpedia. I believe that Semantic Web is the future of web technologies which can lead to a whole new perspective for web usage. Where the fusion of background knowledge with data from event streams can make these streams interpreted by machines knowing the events' relationships to other sources[1] which consequently can lead to huge changes in data processing on web pages. The thing that motivated me to do my research in this field in the context of event streams and implement a useful web application based on Semantic technologies that also utilize big data technologies cloud computing tools. We're interested in utilizing a distributed real-time processing environment as Apache Flink, which is an open source scalable distributed data processing platform that supports batch and Stream processing, in parallelizing the tasks of the application. The aim of this thesis is to distribute the process of enrichment efficiently and consequently build a user-friendly interface to annotate streams and enable the user to apply basic operations on the streams.

Task Build a fully functional open source web application that:

- Allows users to define event sources such as web sites, RSS feeds and push services and to design a Mining Model that specifies the processing steps.
- Enables semantic enrichment by allowing connection to Annotation services such as DBpedia Spotlight to annotate the input stream and display the resulting semantic stream.
- Uses a distributed real-time processing environment, such as Apache Flink, as infrastructure for the application to distribute the tasks and decrease processing time and utilize the Stream processing (DataStream Api) to provide some functionalities that the user can apply on the input streams.
- Allows passing real-time data through the Mining Model and viewing the results.

The system would be useful as it will produce a functional open source web application which applies research results, like those published in[2] and[3] on optimizing the process of annotating different input streams along with integrating Apache Flink to guarantee scalability. Future work might include supporting further annotation services and enabling more operations to be applied on event streams.

3 Acknowledgment

This work has been partially supported by the "InnoProfile-Transfer Corporate Smart Content" project funded by the German Federal Ministry of Education and Research (BMBF) and the BMBF Innovation Initiative for the New German Länder - Entrepreneurial Regions.

References

[1] Charu C. Aggarwal. *Social Network Data Analytics*. Springer Publishing Company, Incorporated, 1st edition, 2011.

[2] Vladimir Batagelj, Anuška Ferligoj, and Patrick Doreian. Special issue on blockmodels direct and indirect methods for structural equivalence. *Social Networks*, 14(1):63 – 90, 1992.

[3] Marc Bertin. Categorizations and annotations of citation in research evaluation. In David Wilson and H. Chad Lane, editors, *Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference, May 15-17, 2008, Coconut Grove, Florida, USA*, pages 456–461. AAAI Press, 2008.

[4] Marc Bertin and Iana Atanassova. Semantic enrichment of scientific publications and metadata: Citation analysis through contextual and cognitive analysis. *D-Lib Magazine*, 18(7/8), 2012.

[5] Michele Coscia, Fosca Giannotti, and Dino Pedreschi. A classification for community discovery methods in complex networks. *CoRR*, abs/1206.3552, 2012.

[6] Julien Desclés, Olfa Makkaoui, and Taouise Hacène. Automatic annotation of speculation in biomedical texts: New perspectives and large-scale evaluation. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, NeSp-NLP '10, pages 32–40, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[7] Wenjie Fu, Le Song, and Eric P. Xing. Dynamic mixed membership blockmodel for evolving networks. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 329–336, New York, NY, USA, 2009. ACM.

[8] Lise Getoor and Ashwin Machanavajjhala. Entity resolution: Theory, practice & open challenges. *Proc. VLDB Endow.*, 5(12):2018–2019, August 2012.

[9] Sean Gilpin, Tina Eliassi-Rad, and Ian N. Davidson. Guided learning for role discovery (GLRD): framework, algorithms, and applications. In Inderjit S. Dhillon, Yehuda Koren, Rayid Ghani, Ted E. Senator, Paul Bradley, Rajesh Parekh, Jingrui He, Robert L. Grossman, and Ramasamy Uthurusamy, editors, *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pages 113–121. ACM, 2013.

[10] Yashodhara V. Haribhakta, Arti Malgaonkar, and Parag Kulkarni. Unsupervised topic detection model and its application in text categorization. In Vidyasagar Potdar and Debajyoti Mukhopadhyay, editors, *CUBE International IT Conference & Exhibition, CUBE '12, Pune, India - September 03 - 06, 2012*, pages 314–319. ACM, 2012.

[11] Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. Rolx: structural role extraction & mining in large graphs. In Qiang Yang, Deepak Agarwal, and Jian Pei, editors, *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 1231–1239. ACM, 2012.

[12] Keith Henderson, Brian Gallagher, Lei Li, Leman Akoglu, Tina Eliassi-Rad, Hanghang Tong, and Christos Faloutsos. It's who you know: graph mining using recursive structural features. In Chid Apté, Joydeep Ghosh, and Padhraic Smyth, editors, *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 663–671. ACM, 2011.

[13] Thomas Olsson. Understanding collective content: purposes, characteristics and collaborative practices. In John M. Carroll, editor, *Proceedings of the Fourth International Conference on Communities and Technologies, C&T 2009, University Park, PA, USA, June 25-27, 2009*, pages 21–30. ACM, 2009.

[14] Ryan A. Rossi and Nesreen K. Ahmed. Role discovery in networks. *CoRR*, abs/1405.7134, 2014.

[15] Ryan A. Rossi, Brian Gallagher, Jennifer Neville, and Keith Henderson. Role-dynamics: Fast mining of large dynamic networks. *CoRR*, abs/1203.2200, 2012.

[16] Ryan A. Rossi, Brian Gallagher, Jennifer Neville, and Keith Henderson. Modeling dynamic behavior in large evolving graphs. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 667–676, New York, NY, USA, 2013. ACM.

[17] Brizan Tansel, David Guy Brizan, and Abdullah Uz Tansel. A survey of entity resolution and record linkage methodologies. *Communications of the IIMA*, pages 41–50, 2006.