Identifying Experts Through a Framework for Knowledge Extraction from Public Online Sources

Anna Hristoskova
Department of Information Technology, Ghent University
G. Crommenlaan 8 (Bus 201)
B-9050 Ghent, Belgium
ahristos@intec.ugent.be

Elena Tsiporkova
ICT & Software Engineering Group, Sirris
A. Reyerslaan 80
B-1030 Brussels, Belgium
elena.tsiporkova@sirris.be

Tom Tourwé
ICT & Software Engineering Group, Sirris
A. Reyerslaan 80
B-1030 Brussels, Belgium
tom.tourwe@sirris.be

Simon Buelens
Department of Information Technology, Ghent University
G. Crommenlaan 8 (Bus 201)
B-9050 Ghent, Belgium
simon.buelens@ugent.be

Mattias Putman
Department of Information Technology, Ghent University
G. Crommenlaan 8 (Bus 201)
B-9050 Ghent, Belgium
mattias.putman@ugent.be

Filip De Turck
Department of Information Technology, Ghent University
G. Crommenlaan 8 (Bus 201)
B-9050 Ghent, Belgium
filip.deturck@intec.ugent.be

ABSTRACT

The paper describes a dynamic framework for the construction and maintenance of an expert-finding repository through the continuous gathering and processing of online information. An initial set of online sources, relevant to the topic of interest, is identified to perform an initial collection of author profiles and publications. The extracted information is used as a seed to further enrich the expert profiles by considering other, potentially complementary, online data sources. The resulting expert repository is represented as a graph, where related author profiles are dynamically clustered together via a complex author disambiguation process leading to continuous merging and splitting of nodes. Validation of the proposed approach shows an improvement of 17% of the results from DBLP.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*clustering, information filtering, relevance feedback*; I.2 [**Artificial Intelligence**]: Learning—*knowledge acquisition*; I.5 [**Pattern Recognition**]: Clustering; E.1 [**Data Structures**]: [graphs and networks]

General Terms

Algorithms, Experimentation, Measurement, Verification

Keywords

author disambiguation, data processing, clustering, graph data model

1. INTRODUCTION

With the rise of the social web and the advent of linked data initiatives, a growing amount of data is becoming publicly available: people interact on social networks, blogs and discussion forums, research events publish their programs including article abstracts and authors, technological conferences and exhibitions advertise new products and technologies, governments and commercial organizations publish data, and the linked open data cloud keeps on growing. An enormous potential exists for exploiting this data by combining it and extracting intelligence.

Leading search engines mainly provide keyword-based results in response of a search query. This is limited in terms of accuracy and efficiency of information comprehension as one still has to manually search for more data on authors, their level of expertise and their connections. Therefore research on identifying experts from online data sources has been gradually gaining interest in the recent years [1, 3, 8, 6]. However, there are several shortcomings associated with the existing approaches: lack of focus on realistic applications, limited to a single source [5], targeting too large scale [7], poor resolution and accuracy, high information redundancy.

The presented paper supports this upcoming research by creating a framework that constructs an expert-finding repository in an incremental fashion through the continuous gathering and processing of user-related information from a variety of online sources. This allows users to query the expert repository with a set of keywords defining the subject area they want to investigate. The outcome is a list of authors, ranked by decreasing level of expertise on the specific subject. Each author is accompanied by a profile, containing a list of papers, highly touted co-authors and any other information the user might find useful. Such profiles are used in many different applications, e.g. the identification of experts in a particular technological domain (for the purpose of technology scouting), the matching of partners for research proposals, or the visualization of research activities and experts within geographical regions (technology brokerage).

The paper starts with defining several research challenges in Section 2. The actual implementation is thoroughly explained in Section 3, which makes use of a graph representation of the expert model. The clustering process, responsible

DIR 2012 23–24 February 2012, Ghent,Belgium

for identifying the author clusters, is one of the key components. The article ends with a comparative analysis of the results in Section 4. Finally, the main conclusions and future improvements are drawn in Section 5.

2. RESEARCH CHALLENGES

Although a large pool of the data, which is required for applications as described above, is available on the web, it is often still gathered manually. This is a time-intensive, tedious and error-prone process due to the fact that the data is not centralized, is available in different formats, can be outdated or contradictory. Most applications that automatically gather user information from the web serve personalization or recommendation purposes. These differ significantly from the wide range of potential applications mentioned in the introduction, which often impose very strict requirements:

- Very *high* (if not complete) *coverage* over the domain should be attained. This requires information extraction from multiple heterogeneous data sources; structured (LinkedIn, Twitter), semi-structured (ACM DL, DBLP) and unstructured (conference pages).
- The data needs to be *up-to-date* at all times resulting in a data streaming pipeline that continuously presents newly gathered information to approve new, or revoke previously taken decisions.
- *High accuracy/reliability* should be guaranteed. This demands the development of advanced disambiguation techniques and the quantification of the different sources in terms of reliability and trustworthiness (e.g. distinguish between doubtful and reputable sources).
- It should be possible to *rank the experts* in terms of impact and relevance. Identification of adequate criteria and metrics, which most probably will be application- and problem-dependent, allowing to perform multi-criteria decision analysis is necessary.

These requirements are taken into account in the next section using a bottom-up approach to building the expert-finding repository.

3. BOTTOM-UP CONSTRUCTION OF AN EXPERT-FINDING REPOSITORY

We propose a bottom-up expert-finding approach, which implements an entity resolution method allowing for reliable disambiguation of authors of scientific articles. Its internal functioning is split up in three main components (Figure 1): *gathering data* from various online sources (publications, author profiles, online presentations), improving accuracy through *data cleaning* and *disambiguation* between authors, and *analyzing* and clustering this data to a specific author. The overall result is defining the areas of interest of each author and their level of expertise for each of them.

The data gathering results in high coverage through the incremental extension of initial seeds. Identified online sources to mine serve as seeds for the incremental growth of the repository, targeted to the application domain in question. This requires web scraping techniques extracting necessary information such as an initial list of authors, article title, abstract, co-authors, affiliation. Using the extracted information additional sources are considered, such as Google Scholar, DBLP, Microsoft Academic Search, in order to search for authors and co-authors and identify additional published material. This results in a broader set of actors in the field, technology-related publications, research activities and author career evolution. The expert repository is dynamically updated with this infinite stream of information.

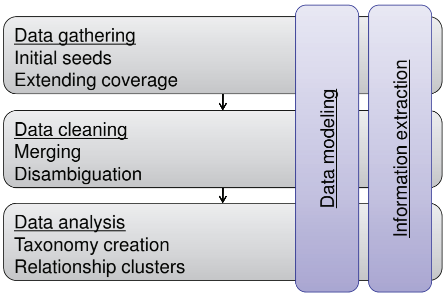

Figure 1: A bottom-up approach to building an expert-finding repository.

The collected repository data consists of partial information on entities (authors) and relationships (links between authors), which are often inconsistent and conflicting. For instance an author's name is not a unique reference to a person as there might be multiple authors with the same name or the name can be spelled differently or change throughout time. It should be possible to discriminate between different individuals with similar names. *Merging* and *disambiguation* are required to guarantee that an expert profile and associated publications refer to a unique author. During the data collection phase each author's name is stored as a new entity in the repository, even if that name is already present. This is necessary in case the same name is connected to different authors. The *disambiguation* of authors consists of a number of rules (detailed in Section 3.2) which inspect several entities in the repository and define the probabilities that names, typically connected to a publication or a profile, represent a unique author. *Merging* clusters the names so they would reference the same author using the probabilities calculated during the disambiguation phase.

3.1 Graphs as a Flexible Data Model

As new information is gathered constantly, the results of the disambiguation and merging phases are not permanent as decisions might need to be revoked. This requires a data model enabling flexible management of the continuous stream of partial information.

The extracted information comprises entities (authors) and relations between them. Creating an ontology with this information allows to represent it and reason about it in a general way independent of the specific domain at hand. As ontologies can be viewed as graph structure the selected representation method is a graph-based model. This graph model is composed of three layers, combining the *structural*, *informational* and *algorithmic* aspects that emerge from dealing with the complexities related to author disambiguation.

The *structural layer* defines the graph structure of authors and reflects the disambiguation decisions through a change in structure. Extracted information is represented as an 'instance' consisting of a collection of nodes and edges that describe (partial) information about an author. Constructing a complete author profile amounts to finding an optimal partitioning (clustering) of instances resulting in each instance-group (cluster) representing a unique author.

The *information layer* comprises the data itself and structures the partial author information. The authors are considered unique containing name instances linked to their publications. There is no limit on the amount of data. Every new addition to the author profile (publications, locations, events) is used to produce similarities increasing the precision of the framework. This data flow is constantly updated.

Finally, the *similarity layer* defines similarities between author instances, performs clustering and links the instances referring to the same author. Every time new similarity is computed, it is possible that reclustering occurs. The constant influx of information requires a dynamic approach, detailed in the next section, that maintains the cluster quality.

3.2 Continuous Incremental Clustering

Similarity edges are added between author's name, e-mail address and affiliation nodes. A domain-independent dynamic minimum-cut tree algorithm described in [4] computes clusters based on these similarity edges. Only part of the minimum-cut tree is build as the number of authors impacted by new data entries is limited and the tree is computed over subset of nodes affecting limited number of clusters. This solution guarantees efficiency while maintaining an identical cluster quality as the static version of the algorithm. The sequential Gusfield's algorithm described in [2] is implemented.

Additionally domain-dependent rules propagate similarities when clustering occurs. They drive the entire flow of the framework by converting new information into similarities between instances. The four rules that are examined are:

Community: exploits the fact that authors often work together with the same co-author. Figure 2 gives a visual representation of how this works.

Interest: the subjects of publications of the same author are usually located within the same field of research. We define keywords extracted from the titles of the papers as the author's interests.

E-mail: authors with the same e-mail address, are most likely the same person.

Affiliation: authors are more likely to work at one affiliation at a given time.

The similarity edge between two instances is assigned specific weight. This weight is calculated by the disambiguator that defines priority weights and thresholds to compute the probability that the parameters (community, interest, e-mail, affiliation) of the author instances match.

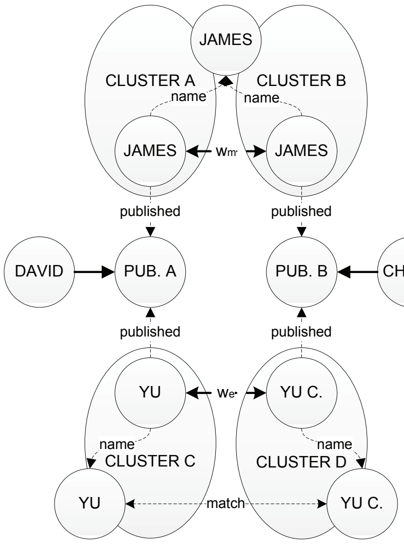

Figure 2: The co-author rule in action: comparing the two instance of James, a similarity (w_m) is added as the co-authors Yu and Yu C. match.

Rules are triggered by different events in the system. A rule is for example executed when it has been discovered that an author has published a new publication, but is also executed on the event of a reclustering. The latter is a by-product of the system itself and not originating from an external source. Rules are performed on three different scopes: instances with the same name, instances with similar names and instances that are part of the same cluster. Strictly respecting this scopes narrows down the problem domain.

The clustering process is implemented as a stateful pipe. It is completely decoupled from the graph representation and is almost not being accessed during the clustering process. The reasoning about the grouping of instances is done completely local and the state of the similarities is maintained in a shared key-value store. This approach takes a lot of the load off the repository, which is important as a graph repository does not scale that easily.

4. INITIAL FRAMEWORK EVALUATION

The clustering solution is evaluated on five family names (Woo, Turck, Mens, Chen and Johnson), each with a number of variations. These were manually disambiguated combining the authors into clusters using the information on DBLP and the actual papers. In total the constructed ground truth test set contains just over 1000 publications. The comparison between the number of authors represented by DBLP and the number of authors we disambiguated, is also presented in Table 1.

The family names of the authors are selected as initial seeds while searching for publications on DBLP. This is combined with e-mail and affiliation information that has been composed manually. The result is a graph containing clusters with the different authors. Next precision, recall and F-measure, as defined in Equation 1, are calculated by comparing the calculated clusters extracted from the graph computed by the minimum-cut tree algorithm and the different

Family name	Manual	Publications	DBLP
Turck	4	172	4
Chen	70	221	1
Woo	1	9	3
Mens	2	153	2
Johnson	107	460	64

Table 1: Comparison between the classification of the manually disambiguated family name dataset and the results from DBLP.

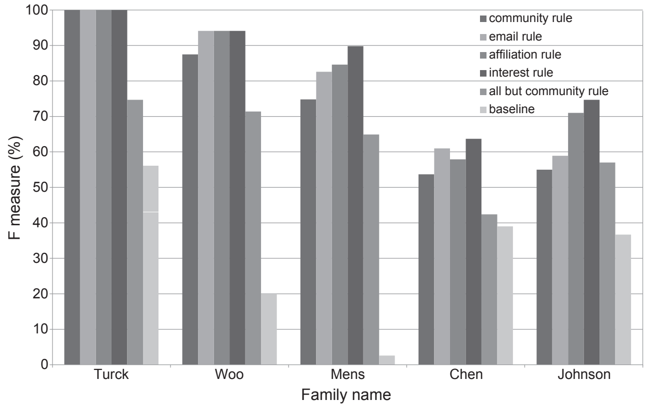

Figure 3: A comparison of different combinations of rules. The first four columns stack the rules, the fifth column uses all rules except the community rule and the last column depicts the base line, this is the F-measure of the case where no clustering has happened.

rules with the manually composed data set.

$$\begin{aligned} precision &= \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} \\ recall &= \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|} \\ F_\beta &= (1+\beta^2) * \frac{precision * recall}{\beta^2 * precision + recall} \end{aligned} \tag{1}$$

The impact of each of the rules on the accuracy is tested for each of the family names. The F-measure for each of these combinations can be seen on Figure 3. The combination of all four rules renders the best result, although sometimes the increase in accuracy from an additional rule is minimal. In the case of "Chen", adding the affiliation rule to the community and e-mail rule even results in a small decrease in accuracy. This is because certain authors are wrongly clustered together. The co-author rule on the other hand has the biggest positive impact on the correctness.

The F-measure for each of the family names as divided on DBLP is also calculated, to make a comparison with the presented results in this paper. Table 2 and Table 3 show that the proposed solution overcomes DBLP by 14% or 17%, depending on how the mean accuracy is calculated.

5. CONCLUSIONS

This paper presents an expert-finding repository focusing on author disambiguation by implementing a dynamic clustering algorithm, allowing for real-time applications. An initial prototype has been developed that continuously gathers data based on initial seeds using a flexible graph as a data model. It incrementally clusters authors based on a domain-independent algorithm and a set of domain-dependent rules. Validation of the proposed approach shows an improvement of 17% of the results from DBLP.

%	Turck	Woo	Mens	Chen	Johnson
DBLP	100.0	87.5	100.0	2.7	62.8
Proposed	100.0	94.1	89.8	63.7	74.7

Table 2: Comparison of the F-measures for the different family names as divided on DBLP and as calculated by the proposed expert repository.

%	Mean	Weighted
DBLP	70.6	61.8
Proposed	84.5	79.0

Table 3: Comparison of the mean F-measure and a weighted distribution based on the number of papers of each author.

Future work should focus on enabling the usage of negative weights to the graph model. The expansion of the number of online sources in order to retrieve more author information will result in the possible entailment of additional (re)clustering rules.

6. REFERENCES

[1] K. Balog and M. de Rijke. Finding similar experts. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 821–822. ACM, 2007.

[2] G. Flake, R. Tarjan, and K. Tsioutsiouliklis. Graph clustering and minimum cut trees. *Internet Mathematics*, 1(4):385–408, 2004.

[3] H. Jung, M. Lee, I. Kang, S. Lee, and W. Sung. Finding topic-centric identified experts based on full text analysis. In *2nd International ExpertFinder Workshop at the 6th International Semantic Web Conference ISWC 2007*, 2007.

[4] B. Saha and P. Mitra. Dynamic algorithm for graph clustering using minimum cut tree. In *Proceedings of the 6th IEEE International Conference on Data Mining ICDMW '06*, pages 667–671. IEEE, 2006.

[5] I. Soboroff, A. de Vries, and N. Craswell. Overview of the trec 2006 enterprise track. *TREC 2006 Working Notes*, 2006.

[6] M. Stankovic, J. Jovanovic, and P. Laublet. Linked Data Metrics for Flexible Expert Search on the Open Web. In *Proceedings of the 8th Extended Semantic Web Conference ESWC 2011*, pages 108–123, 2011.

[7] C. Whitelaw, A. Kehlenbeck, N. Petrovic, and L. Ungar. Web-scale named entity recognition. In *Proceeding of the 17th ACM Conference on information and Knowledge Management*, pages 123–132. ACM, 2008.

[8] J. Zhang, J. Tang, and J. Li. Expert finding in a social network. *Advances in Databases: Concepts, Systems and Applications*, pages 1066–1069, 2010.