Enterprise Collective: Connecting People via Content

Omer BARKOL
Hewlett-Packard
Haifa, 32000, Israel

Ruth BERGMAN
Hewlett-Packard
Haifa, 32000, Israel

Kas KASRAVI
Hewlett-Packard
W. Bloomfield, MI 48323, USA

Shahar GOLAN
Hewlett-Packard
Haifa, 32000, Israel

Marie RISOV
Hewlett-Packard
Pontiac, MI 48341, USA

ABSTRACT

We describe an application for rapidly and optimally responding to enterprise opportunities and challenges, by leveraging the tacit knowledge in an enterprise, via identifying the right subject matter expert(s). Enterprise Collective is a web application that automatically discovers experts and their expertise via semantic analysis of their work products (e.g., e-mails, patents, papers, reports, presentations, and blogs). The key feature of Enterprise Collective is being “passive”; where the employees do not fill out or maintain forms or profiles. The application provides an interactive user interface that hides the underlying complexity. Enterprise Collective can benefit any business user, without extensive training or any analytical background. The application leverages the Expert-Expertise, Expert-Documents, and Expertise-Documents relationships, and subsequently permits navigation within this knowledge space. Enterprise Collective uses technologies for semantic analysis of work products and relevance computation using graph flow. A semi-automatic taxonomy generator is used to extract expertise from documents. The “authority” of each expert in relation to an expertise is computed via the nature of the work product and frequency of references. To demonstrate the benefit of Enterprise Collective in large organization, we describe a case study.

Keywords: Collaboration, People Finding, Text Analysis, Taxonomy Generation.

1. INTRODUCTION

Global competition and accelerating pace of business require rapid response to opportunities or challenges. In most cases, the tacit knowledge of the subject matter experts offers the best means to the right solution (e.g., the explicit answer, documents, or other experts). For our purpose, we define an “expert” as any employee who possesses the required knowledge, and the “expertise” is defined as the required knowledge or skill. Therefore, we define the problem as rapidly finding the expert(s) needed to respond to an opportunity or a challenge. The opportunities may include responding to sales pursuits, and the challenges may include resolving specific business or technical problems. It is also plausible to consider that there are many pockets of expertise that are hidden to a requestor in large enterprises. If not tapped, the value that these hidden experts can generate will be lost.

We form organizations because collectively we can accomplish more than individually. However, it has been suggested that “communication/collaboration within an organization becomes difficult, when an organization reaches about 250 in size” [19]. Traditionally, we have relied on personal networks to develop and maintain a group of experts. During the past decade, many attempts have been made via knowledge management (KM) systems, human resource (HR) forms, online resumes, and profiles to help address this problem. Most of these solutions may be classified as Systems of Records, as described by Geoffrey Moore [7]. Arguably, most of these solutions have failed at being effective at addressing the problem, and not in large-scale use. We argue that the reason is the amount of time it takes for so many people to create and maintain the data in such systems. Enterprise Collective is an application for rapidly and optimally responding to enterprise opportunities and challenges, by leveraging the tacit knowledge in an enterprise, via identifying the right subject matter expert(s). Enterprise Collective is a web application that automatically discovers experts and their expertise via linguistic and semantic analysis of their work products (e.g., e-mails, patents, papers, reports, presentations, and blogs).

The system combines multiple data sources, including unstructured work products and structured sources such as Enterprise Directory and document meta-data, into one holistic model of organizational knowledge. The key feature of Enterprise Collective is being “passive”, where the employees do not fill out or maintain forms or profiles. The approach taken in Enterprise Collective is to construct a graph model of organizational knowledge. Nodes in the graph correspond to meaningful entities, such as documents, people and semantic tags. Nodes in the graph correspond to relations among entities, such as managed-by, author-of, similar-to, etc. Graph models have been used to capture knowledge in large document repositories. Once the graph model is constructed, answering queries over the graph corresponds to flowing relevance over graph edges. Queries may correspond to questions such as "Who is expert in topic X?", "What is the expertise of person P?", "Document D is interesting, what other documents are relevant?", and more. We observe that finding the right expert is the most critical step in addressing an opportunity or a challenge, because with that person we also gain immediate accesses to experiences, tacit knowledge, and introduction to other experts. Enterprise Collective, thus, aims to attain an accurate, high-fidelity profile of employee expertise.

We observe, however, that what people work on gives better insight to what they know about than what they say. We, therefore, avoid the standard, manual approach to obtaining profiles. Rather we make use of work products as a form of implicit profile. The type of content and frequency of citations is an indication of authority in a subject matter. Semantic analysis of work products is used to automatically generate taxonomy of semantic tags as well as identifying similar documents. We take the approach of inserting semantic information in the graph model, rather than building explicit profiles, which is common practice when the user manually enters a profile (e.g., Jive [25]) or a profile is inferred (e.g., Whodini [24]). To show the value of our approach, we have created an instance of Enterprise Collective which includes approximately 10,000 documents and 10,000 technologists. The application is built as a combination of three main components:

1. A back-end responsible for connecting to the different content types, people finder, and taxonomy creation. These components create a large graph of entities that represent their relationships with each other.
2. An analytics engine that answers relevance queries on the entity-relation graph.
3. A front-end that allows a user to both view and navigate a knowledge-base represented by the graph, but also to search the graph for relevant and personalized information.

We report the details of the proof-of-concept application and discuss quantitative and qualitative results of this experiment.

2. RELATED WORK

The practice of recommending contacts is common place today. Facebook and LinkedIn use link analysis over the network of existing connections [13]. A similar approach has been applied to the network that email communications implies [4, 15]. Social networking tools aimed for the enterprise market rely on an explicit, usually manual, profile, (e.g., Jive [25]). Similar tools exist for conference management (see Presdo [22]) and Customer Record Management (CRM) (see Trampoline [23]). Other types of information that have been used to infer an expertise profile include search query logs, (e.g., [14]), document access patterns [15, 18], and document citations [1].

All these types of information indicate what a person is interested in more than what that person knows about. Our approach uses content of work products as the primary evidence of an employee's expertise. Recent work has used the content of emails to automatically generate profiles, (e.g. [24, 2]). The profiles, however, are explicit and subject to user approval. In addition, we consider email to be the least authoritative type of work product.

There is a large body of work, both academic and applied, on semantic analysis of text. Generation of a suitable taxonomy for enterprise-level utilization has been a non-trivial problem. Both manual and automatic approaches have been attempted. On one hand, vocabularies and classifications manually generated by domain experts tend to be very deep and specific, but generally lacking agility due to the need of manual maintenance. On the other hand, automatically-generated taxonomies based on statistical methods are subject to being too generic, and have high degrees of false positives/negatives [16]. Recent hybrid approaches analyses term statistics with respect to a generic taxonomy, such as Wikipedia [8]. We leverage this approach but merge the result with additional enterprise ontology and with enterprise-specific terms that recur in the corpus that do not appear in the generic taxonomy.

Text analysis techniques assess the similarity between documents. Numerous approaches have been investigated to compute documents similarity using keyword vectors [17], latent semantic indexing [5]. Important characteristics of our work is that it's dynamic and lends itself to an enterprise setting in which new documents are continually added and salient concepts shift over time. The approach is closely related to query expansion by pseudo relevance feedback [10].

We model organizational knowledge repositories as a graph. This approach enables us to automatically combine structured and unstructured data from several enterprise sources. A general approach for personalized relevance computation over a graph is the Personalized Page Rank or Random Walk with Restarts [3, 6]. Our graph flow algorithm is structured and allows us to incorporate domain knowledge to direct the flow.

3. ALGORITHMS

Enterprise Collective consumes data from several data sources. Documents are gathered by crawling multiple document repositories, and both unstructured document content and document meta-data are collected. Manual enterprise taxonomies may be mined. In addition, information about employees is obtained from Enterprise Directory, and this information is further used to disambiguate authors of documents.

These data sources are used to build a graph representation of the enterprise knowledge. This graph has nodes that correspond to documents, people and semantic terms, and edges represent authorship, organizational hierarchy, tagging of documents and people, and finally documents similarity. Some relations such as organizational hierarchy and authorship are copied directly from the input source. Relations such as tagging and similarity of documents are extracted with semantic text analysis algorithms.

Taxonomy Creation

Our approach to taxonomy generation consists of two phases, which enable us to combine the best of manual and automatic techniques to taxonomy generation. In the first phase, we apply a generic taxonomy based on Wikipedia articles and hierarchical categories to describe functions of a global enterprise IT organization with its complex collection of concepts, roles, and functions. In a more specific case, we would like to define a corporate-acceptable vocabulary defining "expertise", as well as generate an expertise hierarchy. For the first phase, we leveraged the Document Taxonomy Extractor (DTX) tool [16] that is also a component of the Taxonom.com package. DTX extracts significant phrases from text, validates them against the Wikipedia corpus, and utilizes Wikipedia hierarchy to place extracted terms. For our purposes, two- and three-word noun phrases were extracted.

The second phase is to use the results of a hierarchically generated taxonomy and in each branch select a node that both belongs to the taxonomy and at the same time is commonly used by the members of a given enterprise. This is most important for the areas of common expertise of the enterprise. Therefore, as the second step, we need to decide at what level the node value would make most sense to the enterprise members. Arguably, manual taxonomies developed by the enterprise itself contain the preferred terminology.

Useful enterprise taxonomies include, for example, a product catalog, internal project names, and Intranet tags. These may be manual taxonomies or automatically generated by crawling of the corporate intranet and extracting statistically significant named entities [9].

We also generally favored a higher-level term (in a taxonomy tree) if more than one node per branch was enterprise-friendly. The identified enterprise-friendly nodes become either a basis for a flat taxonomy, or expertise attached to experts, with the option to navigate up and down the hierarchical tree if desired.

Tagging for documents is assessed relative to the generated taxonomy and these inferred document tags are added as edges to the graph model.

Document Similarity

We compute document similarity using a version of query expansion by pseudo relevance feedback [10], an algorithm that has been shown to be effective for search engines. This version computes the expanding terms using a weighted correlation to the top ranked results, using their scores rather than their ranks only.

To compute document similarity, we first perform term-extraction from the input document and then query expansion as specified above. Our current term extraction evaluates the TF-IDF of each term in order to decide on the probability of the term’s importance for the current document (TF – term frequency) and within the entire corpus (IDF – inverse document frequency).

One can apply many other methods to choose the best terms including (e.g., linguistic processing). We then search for documents that are relevant. This search can also be personalized (i.e., the similarity can be relative to the searcher). Our implementation is based on a Lucene [20] inverted index, which supports the direct index as well. This index also allows a personalized document search in addition to computing document similarity.

The document similarity computation is used to add similarity edges between document nodes in the graph model. These edges carry a weight that corresponds to the computed similarity.

Graph Algorithms (Relevance Engine)

Graph algorithms identify sub-graphs of the entity-relation graph that constitutes various personalized views of the data. Thus, we can create an online view that is relevant to the query and personalized for the user. The graph algorithms allow flowing interest from a node to other nodes in the graph. Specifically, to find the relevance graph of a person, flow can be moved from the person thru her organizational hierarchy to the content they create and from there to similar content that is created in different places in the enterprise. This way, relevant people, content, and topics can be explored.

In the same manner, graph algorithms support relevance computation with other nodes as the reference, which enables us to respond to a wide variety of use cases including:

Identify Expert Expertise: When a person is chosen, the topics returned by the most-relevant-graph computation correspond to the expert’s areas of interest or expertise.

Find Experts from Expertise: When expertise is chosen, the people returned by the most-relevant-graph computation are domain experts.

Technologist Network: When a person is chosen, the people returned by the most-relevant-graph computation form a social network of technologists who share knowledge and interests.

Authority Ranking: When the relevance computation considers the source of the knowledge asset and assigns weights to it (e.g., a patent gets a high score, a presentation gets a lower score, and an email gets a very low score). Ranking experts by computed relevance values is used to establish the “authority” or “credibility” of an expert for each expertise. The premise is, in most cases, each type of content has a different weight in indicating the degree of expertise of an expert. Additionally, we take into consideration the frequency. So the resulting sum of the content type points is multiplied by the logarithmic function of the frequency of occurrence of the expertise, to provide a more realistic assessment of ranking.

4. IMPLEMENTATION

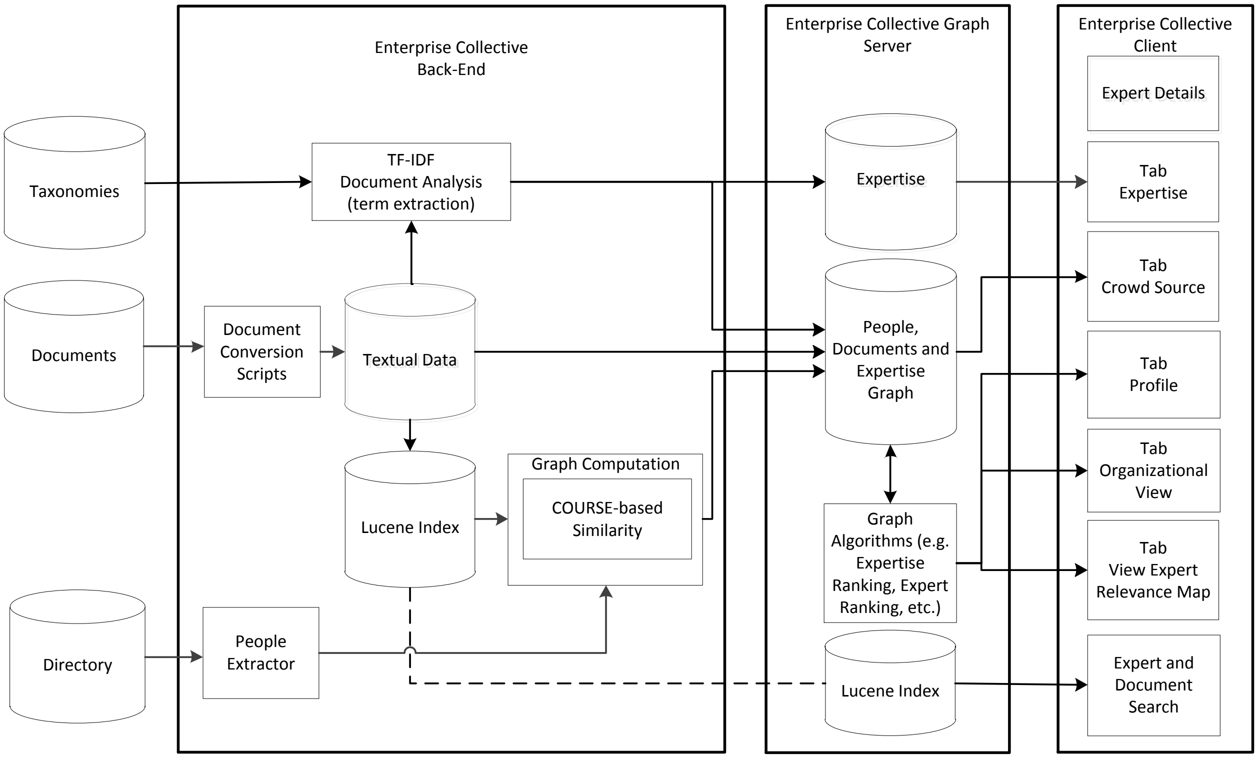
This section describes the implementation of the Enterprise Collective system. A high-level architecture is presented in Figure 1. The key components of the application are:

Data Extractors

The application crawls multiple data sources. The data sources are external to the application and assumed to be constantly updating. Data-extractors connect to data-sources, unify their product, and insert them into intermediate data structures.

Extractors have been implemented for these data sources:

- Document repositories, such as technical reports, patents, and Microsoft SharePoint, which may contain documents, presentations, and emails.
- Enterprise Directory, for information about employees and the organization hierarchy.
- Social networking tools were crawled for content (e.g., blogs, supplementary employee information, in particular, photos).



Enterprise Collective Back-End
Enterprise Collective Graph Server
Enterprise Collective Client
Taxonomies
Documents
Directory
TF-IDF Document Analysis (term extraction)
Document Conversion Scripts
Textual Data
Lucene Index
Graph Computation
COURSE-based Similarity
People Extractor
Expertise
People, Documents and Expertise Graph
Graph Algorithms (e.g. Expertise Ranking, Expert Ranking, etc.)
Lucene Index
Expert Details
Tab Expertise
Tab Crowd Source
Tab Profile
Tab Organizational View
Tab View Expert Relevance Map
Expert and Document Search

Figure 1: Enterprise Collective - High Level Architecture

- Manual taxonomies such as the product catalog and service offerings.
- Internal web was mined for automatic taxonomies.

Back-end Components

The back-end components ingest raw data and, using algorithms described above, construct the graph model.

Graph Model and Graph Server

The graph model is stored in an SQL Server 2008 database. The graph server, which contains the business logic and graph analytics engine, is implemented as a web service on a JBoss 5.1 SOA platform. The service reveals some of the data directly through application programming interfaces (APIs), to allow static information or navigate the entity graph for experts, expertise and clusters of documents.

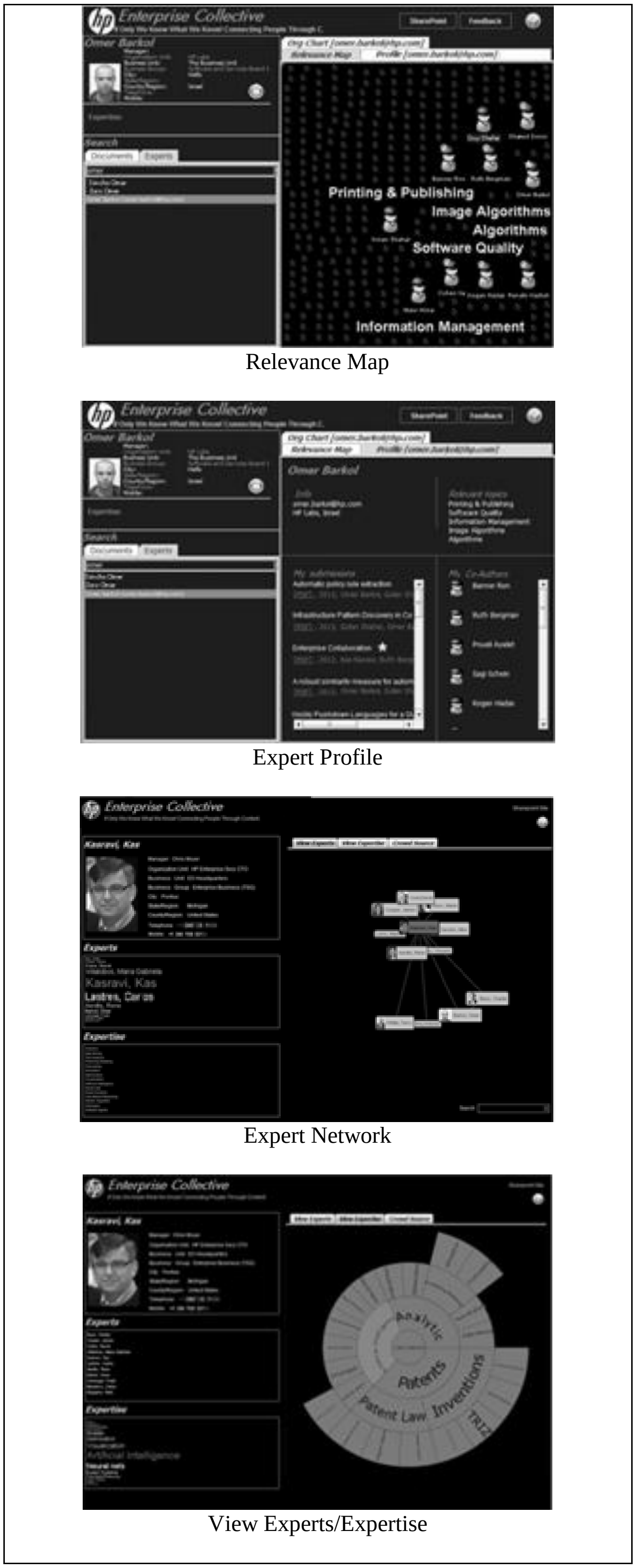
The Client Side – User Functionality

Our premise is convergence of digital documents, Systems of Engagement [7], and analytics serves as the foundation for developing an optimal solution for tapping into an enterprise's tacit knowledge. More specifically, we propose a solution that semantically and passively analyzes the work products of employees, does not require any employee input, and leverages the resulting people connections to help either identify the right expert, or the right group of experts for crowd sourcing. We provide search and navigation capabilities of these experts and their work products to the users.

Enterprise Collective client is a Java applet running in a web browser and is based on Prefuse [21]. The GUI itself is configurable, so different tabs can be exposed or removed per configuration. Figure 2 shows the different capabilities of the user interface of Enterprise Collective.

a) The Relevance Map gives a general view of the experts, documents, and topics that are relevant to the selected person. The visualization allows understanding of the clusters of those resources. It also allows reviewing each of the entities for more information, seeing authorship and co-authorship relations. In all tabs, double-clicking on a person selects this person's node. For documents, one can get the source of the content. For people, by clicking once, one can chose a focus person to get more information in other tabs. Detailed information about this visualization may be found in [12].
b) Expert Profile focuses on the content created by an expert, list of co-authors and other information.
c) The Org-Chart tab, allows the user to see and navigate through his/her organizational hierarchy.
d) Expert Network is another visualization of communities of experts clustered based on expertise.
e) The Expertise tab allows seeing and navigating the different expertise area of an expert. The visualization uses a circular tree structure. This structure also supports moving from expertise to experts and back.
f) The Crowd Source tab allows the user to move between communities of experts to collaborate on content.
g) Expert Details provides general information of the person selected, and can be seen in all the examples in the top left side of the screen shots in Figure 2.
h) The pane on the lower left allows people, expertise, and personalized documents search.

The combination of UI features enables a user to rapidly find the right expert (considering factors such as the level of expertise, organization, and location/time zone). The user can verify the expert's level of expertise through work products in views such as the relevance map or profile. Additionally, a user may discover groups of like-minded people for professional networking, and create optimal groups of people for addressing specific problems.



Relevance Map

Expert Profile

Expert Network

View Experts/Expertise

Figure 2: Sample Screenshots of Enterprise Collective Client

5. EXPERIMENTAL RESULTS – A CASE STUDY

This section describes an instance Enterprise Collective system we have built internally in a corporation of 300,000 people[1]. This instance of the application was made available internally in March 2012. Document repositories were crawled, as described above. The corpora used to construct this instance of Enterprise Collective includes all submissions to an internal technical conference from 2007-2012 (about 10,000 documents), nearly 4,000 patents, 200 technical reports as well as sample documents some participants provided. The application is currently aware of about 10,000 technologists, including authors of documents and their management chain.

Users of the application can:

- View a personalized map of relevant knowledge resources organized around the user's areas of expertise.
- Given a topic, find the most authoritative expert(s).
- Given an expert, identify his/her expertise.
- View a personal profile for every person in the network. The profile includes documents, co-authors, and expertise.
- Explore a hierarchical visualization of the organizational directory to correctly place an unknown technologist.
- Identify the group of experts who would be most appropriate to collaborate on a project or crowd source.
- Assemble a team to respond to RFPs is a use case of particular interest.

The response at the conference was overwhelmingly positive. Users consistently verified the application captured their areas of expertise. They typically recognized most of the people the application identified as relevant. This gave them confidence that the people the application recommends, who they do not know, are very likely relevant as well. We have received feedback from 37 users, 80 percent of whom agree or strongly agree that Enterprise Collective is beneficial. We anticipate considerably greater benefit as we roll out future versions to a greater number of participants.

6. DISCUSSION AND FUTURE WORK

Enterprise Collective is a productivity tool aimed at rapidly responding to opportunities and challenges. The focus of the application is identifying experts and their expertise by using work products generated in their daily activities. We described analytics algorithms, including semantic text analysis and relevance flow by which topics are extracted and expertise profiles are constructed. We built a proof-of concept application of Enterprise Collective, which implements these algorithms and supports several expert and expertise discovery use cases. We are actively researching the graph model and relevance computation using graph flow. We are considering a more structured variant that leverages the structure of the domain. Graph flow can successfully highlight relevant resources when the graph model is set up appropriately. That is, edges truly reflect connections in the organization and the weight of each edge captures the strength of the relation in the real world. At present, the model and connection weights are set in an ad-hoc manner based on trial and error. We seek to replace this ad hoc

[1] The application contains confidential internal content. So, unfortunately, we cannot share it with the reader at this time. We are working on an external instance that will be constructed over academic publications.

process with a rigorous machine learning approach. Our experience has served to emphasize the important of organizational taxonomy, and we have put a great deal of focus on creating taxonomy as described above. We continue to investigate approaches for dynamic taxonomy correction so that our taxonomy adapts as topics shift in the organization.

The next steps for Enterprise Collective include enlarging the scope of the internal proof-of-concept to encompass more employees and additional data sources, both internal and external, (e.g., commonly used social networking data such as LinkedIn connections, email message links, search logs, and document access patterns). We view Enterprise Collective as a production application which will, in due course, be available to enterprises. Enterprises foresee great value in collaboration, which has caused most large organizations to roll out social networking tools on the Intranet. For the most part these tools suffer from lack of adoption, which has been summarized as the 90-9-1 principle – 90 percent of people won't use it, 9 percent will be passive observes and only 1 percent will really engage. We believe that our approach to automatic and explicit profile generation can overcome the adoption barrier by engaging the user from the very first time he or she connects. We are actively building a social networking tool to test this conjecture. Most prior work and expert finding tools use profile similarity. A key technological difference in our approach is that we flow relevance directly between work products, rather than summarizing those in an explicit profile. Taking this approach to the next step, we are considering the use of external search engines directly in the relevance computation so we do not have to explicitly maintain a model containing all the documents. This approach becomes imperative as the population of users and documents increases. It is not yet clear if the current approach scales for very large enterprises. We intend to validate that as we make the application available to a larger install base.

Enterprise Collective investigated some novel rich visualization of relevance recommendations. We believe such rich visualization will be of great value in a research setting. That is, when a person is trying to understand the lay of the land in a new field. Such research is common in enterprises, but no less common in the greater context of the Internet. In future work, we will build a system, using principles and visualizations from Enterprise Collective, to accelerate and simplify the process of researching a new area. Today, the main tool to support such research is a search engine. We, on the other hand, envision search as an underlying building block in our system.

7. ACKNOWLEDGEMENTS

The authors wish to acknowledge the contributions of the following colleagues: Chris Moyer and Dave Cook, for program management; Lior Shapira, Zachi Karni, Reuth Vexler, Meiran Axelrod, and Yaniv Sabo, for analytics engine and visualization research; Bernardo Huberman, for social network ideas; Alexander Ulanov and Natalia Vassilieva, Hui-Man Hou and Shimin Chen, Claudio Bartolini, April Mitchell, and Kivanc Ozonat, and Mike Gardner, for taxonomy; Carlos Lastres and Gaby Villalobos, for development.

8. REFERENCES

[1] A. Balmin, V. Hristidis, Y. Papakonstantinou, "ObjectRank: Authority-Based Keyword Search in Databases", Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004.

[2] K. Erlich, C-Y Lin, V. Griffiths-Fischer, "Searching for experts in the enterprise: combining text and social network analysis", Proceedings of the 2007 international ACM conference on Supporting group work, 2007.

[3] T.H. Haveliwala, "Topic-sensitive PageRank: a context-sensitive ranking algorithm for Web search", IEEE Transactions on Knowledge and Data Engineering, 15(4), pp. 784–796, 2003

[4] I. Guy, M. Jacovi, E. Shahar, N. Meshulam, V. Soroka, S. Farrel, "Harvesting with SONAR – The Value of Aggregating Social Network Information", Proceedings of CHI 2008, April 5-10, Florence, Italy, 2008

[5] T. K. Landauer G. W. Furnas S. C. Deerwester, S. T. Dumais and R. A. Harshman. Indexing by latent semantic analysis. Journal of the American Society of Information Science, 41(6):391–407, 1990.

[6] L. Lovasz, "Random walks on graphs: A survey", Combinatronics (volume 2), pages 1–46, Bolyai Society for Mathematical Studies, 1996

[7] G. Moore. "Systems of Engagement and the Future of Enterprise IT", AIIM, 2011

[8] S.P. Ponzetto and M. Strube, "Deriving a large scale taxonomy from Wikipedia", Proc. of the 22nd national conference on Artificial intelligence - Volume 2 (AAAI), 2007.

[9] K. Ozonat, C. Bartolini, "Automatic Tagging of Documents for the Enterprise Services Sector", International Conference on Artificial Intelligence, in preparation

[10] P. Raghavan, H. Schütze, , C.D. Manning "Introduction to Information Retrieval", Ch. 9, Cambridge Uni. Press, 2008

[11] P. Serdyukov, M. Taylor, V. Vinay, M. Richardson, R. W. White, "Automatic people tagging for expertise profiling in the enterprise". Proceedings of the 33rd European conference on Advances in information retrieval, Springer-Verlag Berlin, Heidelberg, 2011

[12] L. Shapira, Z. Karni, M. Axelrod, S. Golan. "Relevance Maps - Visualization for Recommender Systems", Submitted to InfoVis 2012.

[13] J. Scott, "Social network analysis a Handbook", London, Sage Publications, 1987

[14] Shi, Xiaodong, "Social network analysis of web search engine query logs", School of Information, University of Michigan, Tech Report, 2007.

[15] J.R. Tyler, D.M. Wilkinson, B.A. Huberman," Email as spectroscopy: automated discovery of community structure within organizations, Communities and technologies", Kluwer, B.V., Deventer, The Netherlands, 2003

[16] A. Ulanov, D. Ryashchentsev, "Framework for Effective Representation of Wikipedia and Graph-based Distance Calculation", HP Laboratories Tech Report, HPL-2010-153

[17] A. Wong G. Salton and C. S. Yang. A vector space model for automatic indexing. Communications of the ACM, 18(11):613620, 1975.

[18] F. Wu and B. A. Huberman, "Finding communities in linear time: a physics approach", Eur. Phys. J., B38, 331-338 (2004)

[19] Cambridge Night Interview with Marshall Van Alstyne, Professor at Boston University and Research Scientist at MIT

[20] http://lucene.apache.org/

[21] www.prefuse.org

[22] http://match.presdo.com/

[23] http://www.trampolinesystems.com/

[24] http://www.whodini.com/

[25] http://www.jivesoftware.com/