Crosspoint: A Web-based Collaboration System
Department of Computer Science
Technical Report CS-2009-15

Joel Coffman
Department of Computer Science
University of Virginia
Charlottesville, VA
jcoffman@cs.virginia.edu

Alfred C. Weaver
Department of Computer Science
University of Virginia
Charlottesville, VA
acw@cs.virginia.edu

Abstract

Modern collaboration systems offer users an unprecedented number of ways to communicate but have a critical flaw: they assume users already know with whom they should collaborate. Crosspoint addresses this shortcoming by recommending collaborators and by facilitating collaboration (i.e., setting up collaboration sessions). In this paper, we examine successful social networking sites to identify the plethora of features that users have come to expect from a web-based collaborative environment. We present the design and architecture of Crosspoint and show how it enables expert-finding via a novel search engine that targets the underlying data repository, a relational database. Once a group of experts is identified, Crosspoint facilitates collaboration by hooking into the lightweight (i.e., web) clients provided by third-party collaboration suites.

1. Introduction

Social networking is a concept that has been around much longer than the Internet or even mass communication. People have always been social creatures; some say humanity's greatest asset is the ability to work together to create value that is greater than the sum of the parts. At a minimum, a social network consists of three or more entities communicating and sharing information. This communication centers around any number of socially-constructed groups, the simplest being an organization or university.

Kathleen Carley, a professor of Social and Decision Sciences at Carnegie Mellon University, is one researcher who has studied the importance of social networks and their incredible relevance in people's lives. Following the September 11th attacks, one topic in her studies garnered national attention—the importance of particular entities in a social network. If analysis identified a threat, then the entities should be removed or hindered, thus weakening the organization [1]. Similar information can be used to plan terrorist responses or predict how a corporation will react after a merger [2]. Her research shows that social networks form the foundation of human organization and everything that such organizations produce.

Today, a social network has taken on a popular new meaning. Since the explosion of the Internet age, over one billion people have become connected to the World Wide Web [3], creating an opportunity for communication and collaboration like never before. In today's context, social networking has come to mean individuals using the Internet and web applications to communicate in ways that were previously impossible. The redefinition is largely a result of a culture-wide paradigm shift in the uses and possibilities of the Internet itself.

The mass murder at Virginia Tech on April 16, 2007 provides a poignant lesson concerning social networking. While these tragic events were unfolding during the early morning hours, communication among university officials was understandably conducted via conventional means such as police radio and telephone. As the scope of the tragedy became apparent, notification within the campus (student-to-student and peer-to-peer) was accomplished via email. But for those outside the Virginia Tech community (e.g., students who had friends attending Tech), the status of friends and family was first ascertained via Facebook. Virginia Tech students annotated their personal pages with notes such as "I'm all right," "I'm safe," and "I'm coming home. See you soon."

Although collaboration itself is well-studied, most collaboration systems possess an Achilles' heel: they operate as independent, stand-alone applications with only limited mechanisms for discovery, access, and interaction among users. The tools assume that the user already knows the identities of those with whom he or she wishes to collaborate and requires the user to choose collaborators by name or by group affiliation. This state-of-affairs emphasizes the importance of expert-finding systems when an individual's personal contacts are insufficient for addressing a problem. An expert-finding system augments personal contacts with recommendations derived from data repositories.

We illustrate the importance of these systems through two use cases. First, analysts in the intelligence world are often called upon to answer questions outside of their area of requisite expertise. New analysts initially experience great difficulty because they lack a network of domain experts who assist in solving the problems. This situation limits their effectiveness, even in situations where they are better qualified than the more experienced analysts who merely draw on their larger network of contacts. The medical profession provides our second use case. Although it is unusual for a doctor to see a rare, life-threatening disease, it is paramount that the patient receive the right diagnosis and, if warranted, a referral to a specialist. Once again, both of these objectives are hampered when a doctor lacks a sophisticated network of personal contacts for consultation and referral.

Crosspoint provides a user-friendly environment for problem holders and experts alike. Problems are described from the problem holder's perspective, and the characteristics of the desired outcome are identified. Crosspoint is collaborator-focused, in that users employ a query service to locate experts possessing the required skills and expertise. Search criteria may be prioritized to increase the relevance of the search response. Potential collaborators are known to the system via user-generated profiles, allowing experts to reveal the level of detail with which they are comfortable.

To facilitate rapid adoption, Crosspoint is based upon a Service-Oriented Architecture and is implemented using web services. Both design decisions avoid the creation of yet another "stovepipe" or independent application. The information sharing, collegiality, quick group formation, and sense of shared purpose and shared mission that are the focus of Crosspoint are already present, to greater and lesser degrees, in the various web-based social networking applications in common use.

We begin with a review of modern social networking and existing social networking websites to identify innovative features and then outline the features we incorporated into the Crosspoint system. Section 3 formalizes our requirements and architecture. An overview of the search engine, which is used to identify experts for collaboration, is given in section 4. We end with a brief review of related work in section 5 and our conclusions in section 6.

2. Social Networking

The web as it exists today is surprisingly different than the web of a decade ago. The new focus creates a richer breeding ground for social networking and collaboration. Tim O' Reilly explains the movement of the web in this way:

> The bursting of the dot-com bubble in the fall of 2001 marked a turning point for the web. Many people concluded that the web was over hyped, when in fact bubbles and consequent shakeouts appear to be a common feature of all technological revolutions. Shakeouts typically mark the point at which an ascendant technology is ready to take its place at center stage. The pretenders are given the bum's rush, the real success stories show their strength, and there begins to be an understanding of what separates one from the other. [4]

In an abstract sense, social networking is about everyone. While in the past there was a top-down paradigm of the large media corporations creating content for the common people to consume, the production model has now shifted to where individual users create content for each other. While the means of production were previously concentrated into a few large media organizations, those means are now shared by everyone. The model has changed from top-down to bottom-up creation of content, made possible by new web applications that give power to their users.

2.1. Market Assessment

Before we began our design of Crosspoint, we examined 25 social networking sites and selected several for deeper study. These sites exemplified the fact that a commercially successful social networking site represents a win-win proposition between the user and the provider—namely, both the user and the provider must gain something of value for the site to succeed and grow.

2.1.1. eBay. The world's largest online venue for merchandise posted by both individuals and small businesses, eBay provides the requisite online tools for listing, searching, and payment processing. To build a community of trust, eBay requires both buyers and sellers to register, and sellers must further authenticate their identify by providing financial account information before posting items for sale. Although there was an initial reluctance for a buyer to send money to a seller (without assurance that the seller will fulfill his end of the contract), feedback allows participants to develop an electronic reputation, which serves as a surrogate for trustworthiness.

2.1.2. eHarmony. A sophisticated compatability matching system allows eHarmony to match singles who wish to start romantic relationships. An in-depth questionnaire explores a variety of personality dimensions and serves as the starting point for creating a personality profile. The database of profiles is then searched to determine compatible matches. Users have demonstrated their willingness to provide in-depth, personal, private information provided that their information remains confidential and is used only to enhance the users' end experience.

2.1.3. MySpace. Users create highly-customizable web pages containing pictures, personal information, and multimedia content on MySpace. All information is publicly viewable, but proprietary content is retained by its owner so artists (e.g., bands) can use the site to acquire a fan base.

2.1.4. Facebook. Like MySpace, each user creates a personal profile and then "friends" other users. The standardized user interface makes finding information about others incredibly easy (although Facebook applications allow far greater customization than previously possible) and enables forming groups based on shared interests. The News Feed tool allows users to quickly view any recent updates or developments in the lives of their friends. As previously mentioned, a notable example of Facebook's utility was its role in reporting the status of Virginia Tech students after the April 16, 2007 massacre.

2.1.5. Elance. As an online workspace, Elance connects service providers with customers needing to outsource a project. Service providers bid on projects posted by customers. The customer then reviews the service providers bidding on the project and awards the project to the best choice for the job. The two-stage process (service providers bidding on projects of interest and customers then reviewing the profiles of bidders) ensures an efficient and useful network.

2.1.6. `del.ico.us`. Social bookmarking sites allow users to bookmark and tag web pages. Bookmarks are stored on the `del.ico.us` server and are visible to other users. Tagging allows users to assign their own descriptions to each bookmark, an important feature when searching for content. Because all content is user-generated, the system will—over time—reflect a very accurate characterization of bookmarked pages. Moreover, the server-side software does not require users to install anything before using the service and allows users to retrieve their bookmarks from any computer connected to the Internet.

2.1.7. YouTube. As a free online video sharing network, YouTube allows users to upload videos and share movie clips. Users may browse the videos of other users and recommend popular videos to friends. The popularity and success of YouTube stem directly from two important concepts: the ease of uploading and sharing videos and delegating content policing (e.g., identifying copyrighted and inappropriate material) to users.

2.1.8. Wikipedia. This collaborative online encyclopedia allows anyone to create and to edit the contents of articles. Articles are interconnected using hyperlinks, which allow users to browse related content at the click of a mouse. While any user can propose a change to an article, Wikipedia's strict editorial process allows users to discuss their changes with each other out of view of the public. The separation of the article itself from communication about the article brings stability out of chaos. Editors have the ability to review changes to articles to guard against vandalism and to remove incorrect information.

2.2. Lessons Learned

The bedrock of eBay's success as an online auction site is trust. Effective collaboration also requires trust among the various parties, and we use eBay's surrogate—electronic reputations—for our system. While both eHarmony and Facebook solicit personal information (some of which might be considered private), Facebook allows users to omit information from their profile. For example, the "relationship status" field may be left blank so the individual's profile page will not divulge this information. We adopt this approach because we allow users to search the database directly to locate collaborators. If we made information available to the search engine but not to all other users (e.g., an individual's affiliation with the CIA is marked as private so other users cannot see it when viewing the profile page), search results could subtlety leak the private information (e.g., the query "affiliation:CIA"). We use a standardized interface to display user profiles in an effort to simplify the process of locating relevant details about potential collaborators. Finally, users may easily include multimedia content (e.g., pictures, audio, video, or other files) when describing a problem.

3. Design

In keeping with the approach of `del.icio.us`, Crosspoint is written as server-side software. The user interface is web-based and may be accessed using any browser. This decision aims to ease the strain on system administrators who are already responsible for large toolkits and eliminates the possibility of undesirable interactions with other software packages.

3.1. Requirements

Crosspoint has a Service-Oriented Architecture and is implemented using web services. Industry standard languages and tools were used to avoid dependence on any single product vendor. We briefly describe the major system requirements.

3.1.1. Request for Information Service. End users identify their problems using an electronic form, the Request for Information (RFI). RFIs identify the problem holder and permit specification (via plain text) of the problem description, type of result needed, importance of the request, and a deadline for a response. Each RFI is time-stamped when it is submitted to the Crosspoint database. Analysts can view and modify RFIs to improve their specificity based upon interaction between the analyst and the problem holder.

3.1.2. Subject Matter Expert Profiles. Subject Matter Experts (SMEs) are identified by their electronic profiles. Profiles permit specification of an expert's identity, resources, specialization, and experience. The expert's specialization is identified at two levels: a general characterization of the specialty chosen from a menu and a more detailed specification using plain text. The description of the SME's experience is also given via plain text.

3.1.3. Search Engine. The key element of Crosspoint is the operation and functionality of its search engine. Google's market share speaks volumes about the importance of free-form queries that do not require knowledge of the underlying data sources; we incorporate both aspects into our search engine. When searching the database of RFIs and SMEs, Crosspoint allows search terms to be entered using free text and then uses information retrieval (IR) techniques to rank results. We make two critical observations regarding the search engine. First, existing search engines such as Google cannot be used because the search objects (RFIs and SME profiles) are collections of documents containing links to related information, which may match additional keywords. (By links, we refer to the physical layout of relational databases: foreign keys reference tuples that contain related information.) Second, a Google-style search for the conjunction (AND) of all search terms will not suffice because that approach simply fails when all search terms are not present and leaves the analyst with no search results at all.

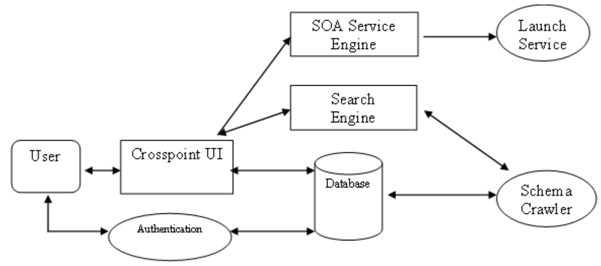

Figure 1. A conceptual diagram of Crosspoint's Service-Oriented Architecture.

3.1.4. Teams. Sometimes no single individual possesses all of the information necessary to respond to a problem. In such cases, a team is formed to facilitate collaboration among the interested parties. Because later problems may be very similar to ones previously handled by some team (or an information need might be recurrent), we retain relevant details so similar problems can be given immediately to an existing team instead of forcing the formation of a new team.

3.1.5. Feedback. When SMEs participate in a team, they have the ability to provide feedback on their collaborators. The feedback ratings form each SME's electronic reputation within the Crosspoint community.

3.2. SOA

Crosspoint's adherence to a Service-Oriented Architecture assures that the services we develop are accessible by any authorized web client, which in turn avoids the generation of a stovepipe application or proprietary designs. Figure 1 shows our SOA, each component of which we describe in detail.

3.2.1. User. The user experiences Crosspoint through a web-based front end. Users are all those who utilize the Crosspoint system, including problem holders, analysts, and subject matter experts. Following the strategy of eBay, we require individuals using the system to first register. When a user submits an RFI, contact information is inferred from the user's profile, thus limiting the amount of additional detail that must be entered. Moreover, the limited information we require during the registration process (i.e., name, email, and password) is no more than the details required to actually contact the user when the problem solution is identified.

3.2.2. User Interface. The web-based UI provides a standardized portal to view information stored within the Crosspoint database and also serves as the gateway to the SOA service engine and search engine. Before gaining access to the information, users must first authenticate. In our prototype, authentication is handled by providing an email address and password, but many other methods of authentication are clearly feasible.

3.2.3. SOA Service Engine. The SOA service engine intermediates between the UI and external collaboration tools (e.g., Microsoft Groove [5] or IBM Sametime [6]). External collaboration environments often provide web-based "lightweight" clients which can be invoked through an ordinary web browser. The SOA service engine maintains separate modules for each external collaboration tool; each module specifies how to create and launch a collaboration session. Naturally, users can opt to store login information for each collaboration tool so that their information will not have to be reentered each time a collaboration session is launched.

3.2.4. Search Engine. Prior to any searches, an external index is created to determine the relationships among the data. The PageRank algorithm is used to determine the prestige of every tuple, which is critical when presented vague (e.g., one word) queries. Users initiate searches through the UI. Queries are parsed into keywords before the full-text indexes of the underlying database are exploited to determine which tuples contain the specified information. The search progresses outwards from the given matches to identify related tuples. For example, an analyst may want to find all SMEs with an expertise in Farsi who are currently in the Middle East. Because this information is stored in multiple tuples, the search engine must recognize relationships in the data. The search expands until the top-k results are identified (where k is the number of desired results) and returned to the user.

3.2.5. Schema Crawler. The schema crawler determines database structure and models the entire database schema as a graph, a requirement for creating the search engine's external index. Each table becomes a vertex in the graph, and edges denote foreign keys between tables. Vertices are decorated with attributes including the names and types of table columns as well as database indexes on the specific table.

Relation	$\text{attribute}_1, \ldots$
User	**id**, name, email
SME	**userId**, specialization, resources, experience
RFI	**id**, *userId*, title, submission, importance, description
Team	**id**, name, description
Affiliation	***teamId***, ***smeId***
Feedback	***teamId***, ***authorId***, ***smeId***, rating, feedback

Figure 2. The schema of the Crosspoint database. Bold text denotes the primary key of each table, *foreign keys* are displayed in italics, and full text indexes are built over underlined attributes.

3.2.6. Database. The Crosspoint database stores all information pertinent to Crosspoint including user contact information, SME profiles, RFIs (both active and archived), and collaboration teams. Any relational database that supports full text search (which is essential for efficient keyword search) may be used; we chose PostgreSQL for our prototype. Core database functionality is abstracted into modules so Crosspoint can easily interface with other relational database platforms (e.g., MySQL or Oracle). The major tables of our schema are shown in figure 2.

4. Search Engine

The Crosspoint search engine incorporates the lessons learned from existing search engines to maximize its effectiveness. As the Internet has proven, keyword search is the medium of choice for both discovering data and retrieving it efficiently per user request. Two important benefits of keyword search stem from what it does not require—namely, a special query language or knowledge of the underlying structure of data.

Unfortunately, an existing search engine cannot be directly integrated despite Crosspoint being a web application. Unlike textual collections of documents, the relational database, which underlies all of Crosspoint, presents a significant challenge to existing search techniques. Searches within a textual document collection (including web pages) typically return the most relevant document(s) (e.g., a single web page) to users. Previous work on structured data sources has defined search results to be a collection of documents, typically joined into a "virtual document," which is most relevant. In the relational model, a collection of documents is essential, because a single data source (e.g., a database tuple) may not contain all the search terms itself but

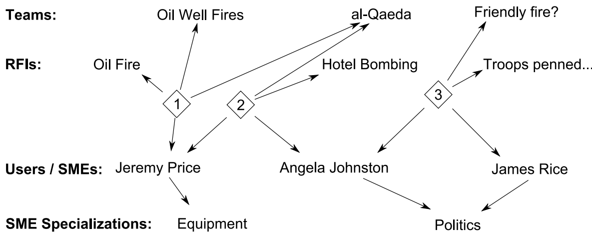

Figure 3. An example data graph for a portion of the Crosspoint database. Each vertex represents a database tuple. Teams appear on the top row, RFIs compose the second row, and SMEs make up the third row of text. SME specializations (e.g., equipment and politics) are found on the final row. Team affiliations are shown by numbered vertices.

may have foreign keys to other tuples which contribute the missing keyword(s). The relational model, which underlies most database management systems in use today, protects data integrity by separating related pieces of information, a process known as data normalization.

In this section, we present an overview of our relational database search engine. Additional details regarding the design and implementation may be found elsewhere [7].

4.1. Query Answers

Previous researchers have defined query answers in two ways. Most systems [8, 9, 10, 11] define answers as trees where every leaf contains at least one query keyword and collectively all of the leaves contain all of the keywords. These answer trees must be minimal: that is, they cannot contain any subtree that satisfies the original definition. One problem with this definition is that it strictly enforces AND semantics although modifications to allow OR semantics have been proposed. We believe a more suitable definition for query answers is individual database tuples, the definition used by ObjectRank [12].

The presence of existing applications for viewing the data stored in the database is our primary reason for selecting ObjectRank's definition. Users are already familiar with navigating among related pieces of information using the preexisting framework, and this framework tags information with user-friendly labels. For example, IMDb (Internet Movie Database) [13] pages link actors to the characters they have played and the films in which they appear. MediaWiki [14] sites (e.g., Wikipedia) are known for their extensive cross references, which link to related content.

The existing browsing framework already provides facilities for navigating among related pieces of information. More importantly, these frameworks links between content that is important to users instead of following every relationship present in the database. Consider the data graph shown in figure 3. One path between the SMEs Angela Johnston and James Rice passes through the politics tuple; this path represents one relationship between the two people. Clearly this relationship is superficial and unlikely to provide great benefit to the user performing the search. If this is the case, the browsing framework will not link an SME to every other expert specializing in politics.

Second, systems that define answers as trees must visualize the individual answer trees (typically as text arranged in a tree or outline view). Although the tree view provides an effective means for understanding the relationships among the keywords, commonalities among the results are easy to overlook. For example, one node might be repeated in many of the top-k answers. In the context of expert-finding, the repetition might suggest an ideal collaborator. Previous research has shown that users process visual cues much more efficiently than text [15], which suggests the best method for displaying the relationships among search results requires a sophisticated user interface whose complexities are beyond the scope of this project.

4.2. Scoring Answers

We use pivoted normalization weighting [16] as the basis of our IR scoring function:

$$\sum_{t \in Q \cap D} \frac{1+\ln(1+\ln(tf))}{(1-s)+s\left(\frac{dl}{avgdl}\right)} \cdot qtf \cdot \ln\left(\frac{N+1}{df}\right) \quad (1)$$

where tf is the term frequency in the document, dl is the document length, $avgdl$ is the average document length for the entire collection, qtf is the frequency of the term in the query, N is the number of documents in the collection, and df is the number of documents containing the term. Pivoted normalization weighting is ideal due to its single tuning parameter, s, for which IR researchers have already identified a good default value (0.2).

Unfortunately, pivoted normalization alone is not sufficient for expressing users' preferences. As previously stated, users prefer results containing all query keywords to be ranked ahead of results containing a subset of the query terms. The document frequency factor of most IR scoring formulas is designed to heavily favor documents that contain terms that occur infrequently in the document collection. Following the work of Luo *et al.* [10], we include a completeness factor so rankings conform to user expectations. Derived from the extended boolean model for information retrieval, the completeness factor is defined as

$$completeness = 1 - \left(\frac{\sum_{t \in Q \cap D} qtw^p (1 - tw)^p}{\sum_{t \in Q \cap D} qtw^p} \right)^{\frac{1}{p}} \quad (2)$$

where qtw is the weight of the query term ($0 < qtw \leq 1$), tw is the weight of the query term in the document ($0 \leq tw \leq 1$), and p is a real-valued parameter in the range $[1, \infty)$. AND semantics are more heavily enforced as the value of p increases. This transition ensures that a document's score reflects user preferences.

Our introductory remarks emphasized that we cannot use existing (e.g., web search) techniques on our database because of the data normalization process. To support relationships (i.e., foreign keys) among the data, we score tuples over a series of time steps. Initially, a tuple is scored according to the attributes it contains; tuples not containing any search terms receive a score of 0 for the first time step. Next, a tuple expands to include the information contained in related tuples—both tuples referencing this tuple and any tuples this tuple references (i.e., edge directionality in the data graph is ignored). The process continues until the tuple exhausts sources of related information.

An external observer might see our scoring strategy like one sees the concentric rings formed when a pebble is thrown into water. The ripple progressively expands until it encompasses the entire body of water or subsides. Similarly, if the data graph is connected, eventually every vertex will be reached. We model the alternative (that is, the ripple eventually runs out of energy and subsides) by introducing a damping factor to control the importance of each successive time step. The damping factor then expresses the user's preference for how closely connected the tuples should be. Our abstraction enables us to use the aforementioned IR scoring formulas without resorting to esoteric adaptations to fit our relational context.

Our model does assume that we can traverse the graph efficiently, which we can do when the entire graph fits in main memory. Alternate designs send complex SQL expressions to the database to discover relationships among tuples. These SQL expressions can be costly for the database to execute. In contrast, we assign every tuple a unique key, which we can quickly derive from the tuple's primary key. We send simple SQL expressions (i.e., those involving a single table) to the database and our graph traversal ensures that we find all related information.

Our in-memory data graph does not track which nodes contain search terms (in contrast to our example in figure 3). Modern relational database management systems feature full-text indexes that quickly retrieve all tuples containing query keywords. Mapping these tuples to their vertex ids has the same effect as including keywords (and their edges) in the graph but clearly reduces the total amount of memory required. Moreover, we reason that a tuple's unique key is far less likely to change than the text contained within the tuple. Thus, we reap the benefit of reusing the existing functionality of the underlying database and our graph remains up-to-date far longer than it would otherwise, provided that the number of insert and delete operations is small compared to the number of updates.

5. Related Work

Expert-finding systems, a generalization of recommender systems, have been extensively studied. McDonald and Ackerman [17] propose an extensible architecture—based on an extensive field study—for locating expertise. Their efforts tease apart the fine line between implementing an expertise recommending system and the formation of social and collaboration networks in real life. Later work by Reichling *et al.* [18] used user-provided documents (in their case, publications) to construct profiles for answering queries. Like ours, their approach is rooted in a variety of information retrieval techniques, but they assume all searchable content is unstructured text. Zhang and Ackerman [19] investigated a number of search heuristics for expertise finding. Our search algorithm is an adaptation of breadth-first search, which guarantees an expert will be found if the social network is connected.

The SmallBlue system [20] has goals similar to Crosspoint's. A number of tools visualize personal networks and locate experts pertaining to a query. Instead of requiring users to maintain profiles, SmallBlue harvests content from email and chat logs while maintaining stringent data-privacy controls. User surveys confirmed the utility of the system and showed that users had no concern regarding privacy issues given the privacy controls in place [21]. Although profiles created from email and chat logs are constantly being updated, both lack fine-grained controls for managing confidential information, which is certainly present in the intelligence domain. User-defined profiles keep users in charge of their personal information and can instantly reflect organizational or job changes instead of having the system slowly adapt over time. Moreover, implementation details (e.g., of the search engine) have not been published, which makes it impossible to compare SmallBlue's search engine with Crosspoint's.

Several different approaches for extending keyword search to relational and XML data have been suggested. BANKS [8] uses backwards-expanding search to approximate group Steiner trees for the data graph. Golenberg *et al.* [11] rank answers by height in order to guarantee efficiency since the group Steiner problem is NP-complete. Both of these approaches duplicate information stored in the database instead of reusing the underlying database's support for full-text search. In contrast to these graph-based algorithms, the work of Hristidis *et al.* [9] and SPARK [10] query a relational database on-the-fly to determine the relationships among keywords. While Hristidis *et al.*'s work was the first to include IR-scoring formulas for ranking results, SPARK adapted pivoted normalization to a relational context and introduces additional normalizations. We have shown elsewhere [7] that our scoring algorithm provides high-quality results, and moreover, it applies to both unstructured and structured text collections.

6. Conclusion

We have presented the design and architecture of Crosspoint, a system targeted at enabling collaboration. The success of existing social networking sites guided our approach and the features included in the system. Crosspoint provides a user-friendly way to request information, a service general to a wide variety of settings. Our search engine fills the void present in today's collaboration systems (that is, no means for identifying collaborators). Together, the services enable Crosspoint to facilitate collaboration, even without providing collaboration tools directly.

7. Acknowledgments

This research was supported by the National Technology Alliance and the National Geospatial-Intelligence Agency under agreement NMA401-02-9-2001/Task No. 01 (IA&E-069). Benjamin B. Morrison contributed significantly to the survey of social networking websites.

References

[1] T. S. Thomas, "Kathleen Carley Garners Lifetime Achievement Award; Her Work 'Enhances Understanding of Social Life'," http://www.cmu.edu/cmnews/011019/011019_carley.html, Carnegie Mellon News, October 2001.

[2] "Dr. Kathleen Carley: Cracking the Foundation," Podcast: http://www.enterpriseleadership.org/listen/podcast-carley, December 2007.

[3] "Usage and Population Statistics," http://www.internetworldstats.com/stats2.htm, Internet World Stats, June 2008.

[4] T. O'Reilly, "What is Web 2.0," http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html, September 2005.

[5] "Microsoft Office Groove," http://office.microsoft.com/en-us/groove/default.aspx.

[6] "IBM Lotus Sametime," http://www-01.ibm.com/software/lotus/sametime/.

[7] J. Coffman and A. C. Weaver, "Effective Searching in Structured Data," submitted to SIGMOD '09.

[8] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan, "Keyword Searching and Browsing in Databases using BANKS," in *ICDE '02*, February 2002, pp. 431–440.

[9] V. Hristidis, L. Gravano, and Y. Papakonstantinou, "Efficient IR-style Keyword Search over Relational Databases," in *VLDB '03*, September 2003, pp. 850–861.

[10] Y. Luo, X. Lin, W. Wang, and X. Zhou, "SPARK: Top-k Keyword Query in Relational Databases," in *SIGMOD '07*, June 2007, pp. 115–126.

[11] K. Golenberg, B. Kimelfeld, and Y. Sagiv, "Keyword Proximity Search in Complex Data Graphs," in *SIGMOD '08*, June 2008, pp. 927–940.

[12] A. Balmin, V. Hristidis, and Y. Papakonstantinou, "ObjectRank: Authority-Based Keyword Search in Databases," in *VLDB '04*, August 2004, pp. 564–575.

[13] "IMDb," http://www.imdb.com.

[14] "MediaWiki," http://www.mediawiki.org/wiki/MediaWiki.

[15] M. M. Sebrechts, J. V. Cugini, S. J. Laskowski, J. Vasilakis, and M. S. Miller, "Visualization of Search Results: A Comparative Evaluation of Text, 2D, and 3D Interfaces," in *SIGIR '99*, August 1999, pp. 3–10.

[16] A. Singhal, "Modern Information Retrieval: A Brief Overview," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, pp. 35–42, December 2001.

[17] D. W. McDonald and M. S. Ackerman, "Expertise Recommender: A Flexible Recommendation System and Architecture," in *CSCW '00*, December 2000, pp. 231–240.

[18] T. Reichling, K. Schubert, and V. Wulf, "Matching Human Actors based on their Texts: Design and Evaluation of an Instance of the *ExpertFinding* Framework," in *GROUP '05*, November 2005, pp. 61–70.

[19] J. Zhang and M. S. Ackerman, "Searching For Expertise in Social Networks: A Simulation of Potential Strategies," in *GROUP '05*, November 2005, pp. 71–80.

[20] C.-Y. Lin, K. Ehrlich, V. Griffiths-Fisher, and C. Desforges, "SmallBlue: People Mining for Expertise Search," *IEEE MultiMedia*, vol. 15, no. 1, pp. 78–84, 2008.

[21] K. Ehrlich, C.-Y. Lin, and V. Griffiths-Fisher, "Searching for Experts in the Enterprise: Combining Text and Social Network Analysis," in *GROUP '07*, November 2007, pp. 117–126.