Extracting social networks and contact information from email and the Web

Aron Culotta, Ron Bekkerman, and Andrew McCallum

Department of Computer Science, University of Massachusetts, Amherst, MA 01003, USA

Abstract. We present an end-to-end system that extracts a user's social network and its members' contact information given the user's email inbox. The system identifies unique people in email, finds their Web presence, and automatically fills the fields of a contact address book using conditional random fields—a type of probabilistic model well-suited for such information extraction tasks. By recursively calling itself on new people discovered on the Web, the system builds a social network with multiple degrees of separation from the user. Additionally, a set of expertise-describing keywords are extracted and associated with each person. We outline the collection of statistical and learning components that enable this system, and present experimental results on the real email of two users; we also present results with a simple method of learning transfer, and discuss the capabilities of the system for address-book population, expert-finding, and social network analysis.

1 Introduction

It is widely held that, while "Internet search" is an extremely important application, "email" is the number one online activity for most users. Despite this, there are surprisingly few advanced email technologies that take advantage of the large amount of information present in a user's inbox.

A person's business effectiveness is often a direct function of his or her ability to leverage the power and expertise of a widely-cast network of acquaintances. Thus electronic address books are increasingly important personal resources for storing contact information of friends, family, and business associates. These address books contain many detailed fields, including street address, phone numbers, homepage URLs, company name, occupation and free-form notes. Recently, some email software also includes relational fields to indicate a social link between two entries. Unfortunately, the task of manually filling in these fields for each entry is tedious and error-prone. One might consider a system that extracts these fields automatically from email messages; however, this approach is limited to the data present in email.

Interestingly, a number of social networking companies have recently been formed to help connect friends and business associates.[1] These companies aim to help business find employees, clients, and business partners by exploiting the topology of their social network. However, the networks these companies search are limited to the people who sign up for the service. Another company[2] extracts university and company affiliations from news articles and Web sites to create databases of people searchable by company, job title, and educational history, but does not address social connections between people.

In light of these partial solutions, this paper describes a powerful, statistics- and learning-based information extraction system for mining both email messages and the Web to automatically extract a user's social network, and to obtain expertise and contact information for each person in the network. After extracting people names from email messages, our system works to find each person's Web presence, and then extract contact information from these pages using conditional random fields (CRFs), a probabilistic model that has performed well on similar language processing tasks [13, 9]. In addition, the system uses an information-theoretic approach to extract keywords for each person that act as a descriptor of his or her expertise.

The system obtains social links by extracting mentions of people from Web pages and creating a link between the owner of the page and the extracted person. The entire system is called recursively on these

[1] http://www.orkut.com, http://www.ryze.com, http://www.linkedin.com

[2] http://www.eliyon.com

newly extracted people, thus building a larger network containing “friends of friends of friends”. This larger network contains a significantly wider array of expertise and influence, and represents the contacts that the user could efficiently make by relying on current acquaintances to provide introductions.

The system provides capabilities and infrastructure for (a) avoiding tedious form-entry of contact and social-linkage information, (b) finding experts in commercial companies or scientific communities, (c) automatically recommending additional experts to CC in an email message about a particular topic, (d) analyzing the social network graph to find the individuals who are hubs and authorities in a particular sub-field, as well as (e) finding the best path between the user and a desired business or social contact, (f) finding communities with high or low connectivity, (g) clustering people both by their attributes and by graph connectivity.

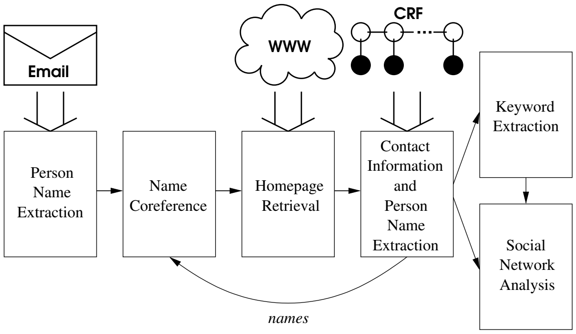

Fig. 1. System overview

We demonstrate our system with experimental results on two real-world email datasets from participants in the CALO project [12]. The primary contribution of this paper is to identify the important components of such a system, describe the performance of our first versions of those components, and identify further areas of research opportunity. In addition, on the application of contact information extraction, we describe and give positive experimental results for a simple method of learning transfer—using labeled data from one task to improve performance on another.

2 System overview

The system's input is the set of email messages in a user's inbox. The output is an automatically-filled address book of people and their contact information, with keywords describing each person, and links between people defining the user's social network. The six modules of the system are depicted in Figure 1 and briefly outlined below.

1. **Person name extraction.** Names are extracted from the headers of email messages by first locating the header of the email, and then using a set of patterns to find people's names and email addresses.
2. **Name coreference.** A set of string matching rules are used to resolve multiple mentions of the same person. For example, we create rules that will merge people with the names “Joseph Conrad” and “J Conrad”.
3. **Homepage retrieval.** We find a person's homepage by creating queries based on the person's name and likely domain. We then submit the query to the Google search engine and filter results based on URL features as well as word distribution similarity metrics between the proposed homepage and email messages (or Web pages) in which we first found this person.
4. **Contact information and person name extraction.** We employ a probabilistic information extraction model to find contact information and person names in the homepages found in the previous step. Newly extracted people who are coreferent to people we have already discovered are resolved as in step 2. Links are placed in the social network between a person and the owner of the web page on which the person is discovered. Additional extraction from the body of the email is an area of ongoing work.
5. **Expertise keyword extraction.** We create keywords for each person by identifying the terms from each person's homepage that have the highest information gain in separating that person from the others.
6. **Social network analysis.** The resulting social network is clustered into communities by a graph partitioning algorithm that searches for and removes edges that connect two highly-connected components.

The following four sections describe in more detail the techniques employed for finding homepages, extracting contact information, extracting keywords, and clustering the resulting network.

3 Homepage retrieval

Given a person's name and possibly an email address, the system finds a person's homepage by the following steps:

1. **Language model construction.** We build a term-frequency-based language model for the target person, from either the email or Web page text in which the person is first seen.
2. **Query generation and Web search.** We generate an ordered list of Web queries from most specific to most general. The first query uses the most common form of the name and uses Google's *site:* operator with the most specific domain name appearing in their corresponding email address or Web page domain (for example *"Tom Mitchell" site:cs.cmu.edu*). If no hits are returned, increasingly general queries are issued; for example, using only the last two components of the domain name (*"Tom Mitchell" site:cmu.edu*). The results of the first query to yield non-empty results are passed to the URL-filtering stage.
3. **URL filtering.** To determine if the URL is a homepage, we note that almost every homepage URL contains some version of the person's name. We apply a string kernel distance measure [11] between various forms of the person's name and each field of the URL, and accept homepages that exceed a threshold. These name forms include the full name, first only, last only, and the email login name (if available).
4. **Homepage retrieval.** For each page that passes this filtering, we crawl its internal hyperlink structure to retrieve the user's entire Web home directory. This results in a larger representation of the person's web presence, and frequently provides pages on which contact information and other people's names are located. To prevent overflow, we limit the total number of retrieved pages per person by a reasonably small constant (e.g. 30).
5. **Filtering irrelevant homepages.** If the site is retrieved in response to a query that includes the Internet domain name, then we conclude that the homepage belongs to the person we are looking for, and omit this stage. However, if the query did not include the domain, we compare the word distribution on this homepage site with the language model constructed in step 1. We currently make the comparison with a threshold on cosine similarity between vectors of word counts.

4 Extracting contact information

To extract contact information from Web pages, we apply a corpus-based, machine learning approach. We first label a training set of documents with 25 fields[3] present in most electronic address books. We then train a linear-chain conditional random field (CRF) to obtain a Markov model for extracting these fields (see [10] for more details). Note that the labels FirstName, MiddleName and LastName are among the labels predicted by the CRF when extracting contact fields, allowing us to extract mentions of other people.

4.1 Learning Transfer

When labeled training data for one task is scarce, it may be desirable to augment it with labeled data or an existing model for some different but related task, thereby "transferring" knowledge from one task to another. This idea is also at the heart of multi-task learning [3], life-long learning [22] and shrinkage [19, 14].

We have currently labeled only 39k words of email and Web data for training a contact extraction system (which is currently a significant cause of limited accuracy). However, we have obtained 1 million words of newswire text labeled for the named entity extraction (NER) task. This newswire data is related to our contact extraction task in that it includes labels for relevant entities such as people, locations and organizations; it is different in that it is missing labels for the majority of our fields (such as JobTitle), in that some of its labels are overly general (such as Person instead of FirstName, MiddleName, LastName), and in

[3] The 25 fields are FirstName, MiddleName, LastName, NickName, Suffix, Title, JobTitle, CompanyName, Department, AddressLine, City1, City2, State, Country, PostalCode, HomePhone, Fax, CompanyPhone, DirectCompanyPhone, Mobile, Pager, VoiceMail, URL, Email, InstantMessage.

that it has some fields not relevant to our task (such as DATE). However, given the overlap, we may hope that information about NER labels would be useful for the contact information extractor.

In this paper we experiment with a straightforward approach: first, we train a CRF to perform the NER task on a large corpus of newswire articles. Then, we train another CRF for contact information extraction using as additional features the labels predicted by the NER model. In cases in which the first model has significant uncertainty, preserving that uncertainty via factorial models [20] may be helpful.

5 Extracting keywords

To associate a set of keywords with each person, we calculate the Information Gain for the terms in each person's web page. We calculate the Information Gain of a person p and a term t as follows: let $X_p \in \{0, 1\}$ be a binary random variable denoting the event of picking person p from the set of all the people. Let $X_t \in \{0, 1\}$ be a binary random variable denoting the event of term t occurring in a randomly chosen document. The Information Gain of X_p and X_t is then defined as $I(X_p, X_t) = \sum_{X_p, X_t \in \{0,1\}} P(X_p, X_t) \log \frac{P(X_p, X_t)}{P(X_p)P(X_t)}$.

For each person, we first build a list of all the terms (unigrams and bigrams with stopwords removed) that occur on the person's homepage, and then we calculate the Information Gain value for each person-term pair. We sort the lists of terms according to their Information Gain values so that the most "meaningful" terms are ranked at the top of the lists, and then prune the lists to the top few terms.

6 Clustering

Since the edges in the extracted social network currently have no attributes—not even weights—we base our clustering algorithm on the work of [23]. This algorithm relies on the notion of *betweenness centrality* [8]. Given all shortest paths between all vertices, the betweenness of an edge is the number of shortest paths that traverse it. The idea is that edges of high betweenness connect people from two distinct communities, while edges of low betweenness connect people within one community.

To efficiently calculate the betweenness of every edge, we rely on a variant of all-pairs-shortest-path, augmented to keep usage counts for edges [2]. It proceeds by looping over each vertex v_i and performing Dijkstra's shortest path algorithm [5] to calculate the shortest between v_i and all other vertices. The betweenness count for each edge (using only the shortest paths for v_i) is calculated and added to a running total. After each v_i has been considered, the betweenness of each edge is exactly half its running total, since we have considered each pair of endpoints twice.

To cluster the graph, we repeatedly remove the edge with highest betweenness until certain stopping criteria are met [23]. After removing each edge, we must recalculate the betweenness of all edges in the connected component from which the edge was chosen, since the removal of an edge of high betweenness greatly affects the resulting set of shortest paths. The running time for a network with more than 3,400 nodes is under two hours.

7 Experiments

We apply our system to the email messages of two participants in the CALO project. Most of the email messages are correspondence between CALO participants on issues related to the CALO project. The data for the first user (**user1**) contains 664 messages. After extracting people names from email headers and resolving coreferent mentions, we obtain 53 individuals, excluding the user. The data for the second user (**user2**) contains 777 messages from which 49 individuals are extracted.

7.1 Homepage retrieval results

We report homepage retrieval results on the **user1** dataset, with two iterations of our system. The cosine similarity threshold is set to 0.1. The system finds the web presence of 31 out of 53 email correspondents

	Token Acc.	F1	Precision	Recall
CRF	94.24	79.70	86.49	73.90
CRF+NER	94.50	80.76	85.73	76.33

Table 1. Token accuracy and field performance for the original CRF and the CRF trained with the output of a named-entity extractor.

and retrieves 229 homepages of people listed on the correspondents' homepages, resulting in 260 retrieved homepages within two degrees of the email inbox owner. We manually evaluated all these sites, looking for the following three types of undesirable cases:

1. **People who are not in the user's social network.** We found 16 instances of this type: 7 due to named entity extraction errors (e.g. *U. S. Healthcare* appearing as a person name), and 4 due to problems of name coreference. The remaining 5 errors of this type occur in homepages that mention other people who are not in the owner's social network, such as novelist *Jane Urquhart.* The latter type of error is especially hard to recognize, but is largely addressed in a subsequent stage that separates these people from the user's community by performing clustering in the social network.
2. **Unrelated people with the same name.** We found 25 errors of this type. Many of these pages relate to other researchers, sometimes to computer scientists—and in some cases it was even challenging for us to manually determine that the found homepage was that of a different person. Other errors are caused by the fact that the cosine similarity of some significantly unrelated pages occasionally surpassed the 0.1 threshold.
3. **Relevant page, but not the homepage.** We found 19 people of this type, two of which do not actually maintain a homepage.[4] Most of these mistakes are the result of Web search and URL filtering stages of the homepage retrieval procedure. In some cases a page dedicated to the person was found on the desired domain, however, his or her actual homepage was on another domain. In other cases the person's username had little in common with the person's real name, so the homepage URL was filtered out.

Thus, not taking into account the first type of anomaly (since it is not a homepage retrieval problem), the precision of finding the relevant person[5] for **user1** dataset is 89.9% and the precision of finding the correct homepage is 82.8%. An important result is that if we exclude the stage 5 filter (see Section 3), precision drops to 47.6%, more than a 35% absolute decrease. We note, however, that the filtering does decrease recall from 283 to 202 pages retrieved.

7.2 Contact information extraction results

To train the CRF, we collected and annotated 2279 files with 25 classes of data fields from various Web and email data, resulting in 26,919 labeled field mentions. About half of the data were isolated address blocks, while the other half were entire Web pages. For testing, we labeled 20 Web pages containing 867 field mentions.

The features consist of the token text, capitalization features, 31 regular expressions over the token text (e.g. ContainsHyphen, ContainsDigits, etc.), lexicon memberships, and offsets of these features within a window of the two previous and the two succeeding tokens. We use 25 lexicons, including lists of popular names, cities, companies, job titles, honorifics, and streets. Some of these lexicons were generated automatically by the information extraction system KnowItAll [7].

As noted in the Learning Transfer section, we also train a CRF which uses the output of a NER system as additional features. We refer to this model as CRF+NER. Table 1 displays the per-token accuracy as well as overall field segmentation performance for the two models.

Note that CRF+NER provides a significant boost in recall. Examining results by field, we notice that CRF+NER improves precision considerably for City (31% improvement), DirectPhoneNumber (+22%),

[4] We do not consider these two people as erroneous cases, so we say the total number of errors of this type is 17.

[5] This ignores both the first and third error types.

Researcher	***Keywords***
William Cohen	logic programming, text categorization, data integration, rule learning
Daphne Koller	bayesian networks, relational models, probabilistic models, hidden variables
Andrew McCallum	information extraction, document classification, language processing, natural language
Deborah McGuinness	semantic web, description logics, knowledge representation, ontologies
Tom Mitchell	machine learning, cognitive states , learning apprentice, artificial intelligence

Table 2. Keywords extracted for some people in **user1**'s social network.

user	$\|V\|$	$\|E\|$	max degree	largest component
user1	3377	3402	315	2827
user2	2019	2027	363	2019

Table 3. Statistics of the clusters for two users. $|V|$ is the number of vertices, $|E|$ is the number of edges, *max degree* is the highest degree of any vertex, and *largest component* is the number of vertices in the largest connected component.

and FaxNumber (+37%), while actually giving worse performance on Country (-7%) and MiddleName (-4%). The improvement for City makes sense given that NER system labels cities as locations and thus provide a useful feature to CRF+NER. However it is unclear why performance increases for phone number fields.

We perform a paired sign test for token-level accuracy and recall. We find that CRF+NER achieves better token accuracy with significance level $p < 0.13$ (insignificant) and better recall with significance level $p < 1.26\text{e-}6$ (significant).

7.3 Keyword extraction and Social Network Analysis

Applying Information Gain to the problem of extracting keywords for people in the user's social network led to highly informative results. In Table 2, for five well established artificial intelligence researchers who are in **user1**'s social network, we list a few highlighting keywords that fell within the top 10 keywords for each person.

Table 3 describes the social networks extracted for each user. Note that in addition to the person links extracted from the Web, we also include links between people who are co-recipients of the same email message. Figure 2 plots the degree distribution of the network, indicating its scale-free properties. Figure 3 displays the clusters discovered for **user2**'s network, within three degrees from **user2**.

While evaluating community discovery is an arguably subjective task, we do find intuitively pleasing hubs (e.g. Michael Jordan of UC Berkeley at the center of a machine learning community), and cluster separation between scientific researchers and celebrities. Many clusters consist of researchers at the same university or in the same field (e.g. we found clusters of researchers in astrophysics, information retrieval, and machine learning, as well as University of Massachusetts faculty). It is noteworthy that many of the hubs of these communities are *not* present in the user's email. This provides an example of helping people locate well-connected contacts in communities of interest. Since the system can maintain the original graph structure prior to clustering, the user can recover the shortest path of introductions to make this valuable connection.

Community discovery is a step toward relation finding because clusters often coincide with a relationship to the user. For example, clusters of people working at the same institution as the user can be labeled "people I work with." Automatically generating such labels is a subject of ongoing work.

8 Related Work

While there has been much work in information extraction and social network analysis independently, we believe this is the first paper to propose an end-to-end system that integrates them both, employing the Web to find information about people in a user's email, and extracting both the user's social network and the contact information for people in this network.

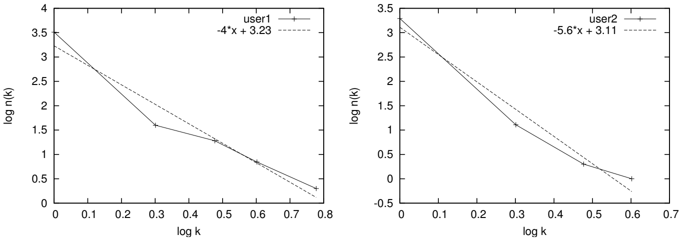

Fig. 2. Log-log degree distribution for the social networks extracted for **user1** and **user2**, where k is the degree and $n(k)$ is the number of nodes with degree k. Note that these distributions approximately follow a Zipf distribution.

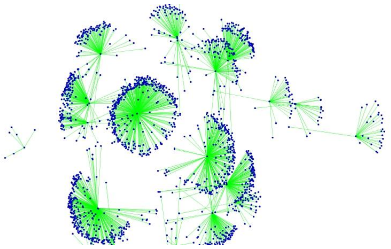

Fig. 3. Clustered social network for **user2**.

1. **Homepage retrieval.** Early work on homepage retrieval in the *Ahoy* system [17] primarily applies a useful collection of heuristic techniques. Xi et al. take a machine learning approach to homepage retrieval using decision trees and logistic regression, however with moderately low results [26]. There also exist information retrieval techniques which augment content-based retrieval with link analysis (e.g. PageRank) and URL features [24]. Since Google incorporates this link analysis into its search, we are effectively employing a PageRank-based filter in our homepage finder.
2. **Contact information extraction.** Previous work on contact record extraction [1] obtains high accuracy using an HMM on a significantly simpler and more limited set of fields (HouseNumber, PO Box, Road, City, State, ZIP), which usually appear in very regular form.
3. **Social network analysis.** Similar social network research is conducted on Usenet data [18], in which the goal is to characterize a dynamic online community as well as determine the "authority" of an individual based on posting patterns. Van Alstyne and Zhang [25] analyze the social network of an email graph; however, their approach assumes access to all email messages, not just those of a single user. While this may be practical for a large company, it is infeasible for the ordinary user.
4. **Clustering methods.** There are many approaches to clustering, including spectral clustering [15], probabilistic relational models [21], graph partitioning [6], and probabilistic latent semantic indexing [4]. The betweenness clustering approach we use here can be viewed as a type graph partitioning algorithm.

9 Conclusions

Email is the primary way that people access their wide-spread social networks. This paper has presented an end-to-end system that automatically integrates both email data and Web content to help users maintain large contact databases, leverage their social network, perform expert finding, and make new relevant connections. The information gathered by the system could also be used as aids to other email functionality, such as automatic foldering and spam detection.

Acknowledgments

We wish to thank Stephen Soderland and the KnowItAll project for providing lexicons. This work was supported in part by the Center for Intelligent Information Retrieval, the Central Intelligence Agency, the National Security Agency, the National Science Foundation under NSF grant #IIS-0326249, and by the Defense Advanced Research Projects Agency, through the Department of the Interior, NBC, Acquisition Services Division, under contract #NBCHD030010. Ron thanks his wife Anna for constant support.

References

1. Vinajak R. Borkar, Kaustubh Deshmukh, and Sunita Sarawagi. Automatically extracting structure from free text addresses. In *Bulletin of the IEEE Computer Society Technical committe on Data Engineering*. IEEE, 2000.

2. U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.
3. Rich Caruana. Multitask learning. *Machine Learning Journal*, 28(1), 1997.
4. David Cohn and Thomas Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *Neural Information Processing Systems 13*, 2001.
5. E. W. Dijkstra. A note on two problems in connection with graphs. *Numerische Math*, 1:269–271, 1959.
6. Chris H. Q. Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings of ICDM 2001*, pages 107–114, 2001.
7. Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Methods for domain-indepedent information extraction from the web: An experimental comparison. In *Proceedings of AAAI*, 2004.
8. L. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41, 1977.
9. Trausti Kristjansson, Aron Culotta, Paul Viola, and Andrew McCallum. Interactive information extraction with constrained conditional random fields. In *AAAI*, 2004.
10. John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.
11. H. Lodhi, J. Shawe-Taylor, N. Cristianini, and C.J.C.H. Watkins. Text classification using string kernels. In *Advances in Neural Information Processing Systems (NIPS)*, pages 563–569, 2000.
12. Bill Mark and Ray Perrault. CALO: a cognitive agent that learns and organizes, 2004. https://www.calo.sri.com.
13. Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In Marti Hearst and Mari Ostendorf, editors, *HLT-NAACL*, Edmonton, Alberta, Canada, 2003. Association for Computational Linguistics.
14. Andrew K. McCallum, Ronald Rosenfeld, Tom M. Mitchell, and Andrew Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In Jude W. Shavlik, editor, *Proceedings of ICML-98, 15th International Conference on Machine Learning*, pages 359–367, Madison, US, 1998. Morgan Kaufmann Publishers, San Francisco, US.
15. A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Neural Information Processing Systems 14*, 2001.
16. Fernando Pereira and Michael Riley. Speech recognition by composition of weighted finite automata. In Emmanuel Roche and Yves Schabes, editors, *Finite-State language processing*, pages 431–453. MIT Press, 1997.
17. J. Shakes, M. Langheinrich, and O. Etzioni. Dynamic reference sifting: A case study in the homepage domain. In *Proceedings of the 6th World Wide Web Conference*, 1997.
18. Marc Smith. Invisible crowds in cyberspace: Measuring and mapping the social structure of usenet. In Marc Smith and Peter Kollock, editors, *Communities in Cyberspace*. Routledge Press, 1999.
19. C. Stein. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability 1*, pages 197–206. University of California Press, 1955.
20. Charles Sutton, Khasayar Rohanimanesh, and Andrew McCallum. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *Proceedings of ICML*, 2004.
21. Benjamin Taskar, Eran Segal, and Daphne Koller. Probabilistic classification and clustering in relational data. In Bernhard Nebel, editor, *Proceeding of IJCAI-01, 17th International Joint Conference on Artificial Intelligence*, pages 870–878, Seattle, US, 2001.
22. S. Thrun. *Explanation-Based Neural Network Learning: A Lifelong Learning Approach*. Kluwer Academic Publishers, Boston, MA, 1996.
23. Joshua R. Tyler, Dennis M. Wilkinson, and Bernardo A. Huberman. Email as spectroscopy: Automated discovery of community structure within organizations. Technical report, Hewlett-Packard Labs, 2003.
24. T. Upstill, N. Craswell, and D. Hawking. Query-independent evidence in home page finding. In *ACM Transactions On Information Systems*, 2003.
25. Marshall van Alstyne and Jun Zhang. Emailnet: A system for automatically mining social networks from organizational email communication. In *NAACSOS2003*, 2003.
26. W. Xi, E. A. Fox, J. Shu, and R. Tan. Machine learning approach for homepage finding task. In *Proceeding of the 9th International Symposium on String Processing and Information Retrieval*, pages 145–159, 2002.