PopulusLog: People Information Database

Ali Cakmak*
{firstname.lastname}
@case.edu

Mustafa Kirac*
{firstname.lastname}
@case.edu

Gultekin Ozsoyoglu
{firstname.lastname}
@case.edu

Department of Electrical Engineering and Computer Science
Case Western Reserve University
Cleveland, OH 44106, USA

Abstract—***Information about individuals on publicly available web sites stands as a valuable, yet unorganized, data source. Turning such an enormous data source into a "database" is highly desirable as it has the potential to lead to novel ways of using the available information to the largest extent. In this paper, we present PopulusLog, a novel web data mining system. PopulusLog is a pioneering example of next generation search engines which produces and provides access to non-intuitive knowledge on the web. It involves a framework for tools that collect, extract, mine, query, browse, and visualize information about anonymous people.***

Keywords— information extraction; search engines; machine learning; web databases; entity tagging.

I. INTRODUCTION

The amount of knowledge available on publicly accessible web pages has been constantly increasing [22]. Among many others, personal information about individuals is one of the most commonly published data type on the web. Politicians, scientists, students, and individuals from different backgrounds publish information about their work, research, experience, family, and so on. This phenomenon has become so commonplace that many people first search (or "Google") a person through a search engine to learn about him/her once they somehow hear about the person. It is also the case that usually information about a person is scattered on many different and possibly unrelated pages, such as a personal home page, an organization's employee profile page, or even a fan club membership page. In addition, most of the time, first and/or last name of a person is not enough to identify him/her uniquely in the space of people who have at least some information available on the web. In such cases, the situation becomes even more complicated as a query with a person name will output all of the information that belongs to people with the given name as if they represent a single person. Moreover, generic search engines do not provide more than links to web pages that contain an occurrence of the given search phrase. Thus, the results need to be aggregated manually. Hence, the generic search scheme misses the useful information that can be obtained only through considering the entities organized as a network. Due to individual consideration of entities, traditional search engines also do not support complex graph queries, such as virtual distance between people.

In order to organize the publicly available personal information on the web in a more structured way, and allow for advanced querying of the collected information, we have been developing a knowledgebase, called PopulusLog [1]. PopulusLog is comprised of multiple tools providing the capability for crawling, information extraction, data mining, and advanced querying. More specifically, PopulusLog (a) allows grouping of (and provides one-point access to) the information about a person, (b) provides semantic querying schemes like "Find the colleagues of person A" rather than just simple syntactical keyword search, (c) evaluates the collected information thorough social network analysis, and provides new knowledge like personal impact factors, social cliques that would otherwise stay implicit, and (d) visualizes the query results. PopulusLog's novel approach to process, organize, and present information about individuals on the web frontiers a new generation of search engine concept with value-added services and capabilities. PopulusLog houses and illustrates a number of such functionalities, e.g., social impact factor calculations, person-to-person similarity searches, personal homepage identification, social network construction, reference resource pages for an individual, and so on.

PopulusLog employs (i) a number of supervised and unsupervised machine learning and data mining techniques (e.g., Support Vector Machines [23], frequent itemset mining [24], etc.), as well as rule-based heuristic methods, (ii) integration of multiple evidence from diverse sources, (iii) client-side lightweight visualization, (iv) confidence scores for the presented information, (v) natural language processing for entity tagging, (vi) advanced graph querying, and (vii) information editing and authentication services.

As a proof-of-concept, we have successfully built a first running version [1] of PopulusLog on Case Western Reserve University (CWRU or Case) web domain, and we describe its features on the data collected from this network. However, PopulusLog can be directly adapted to work on the whole web with minimal further effort. Presently, the PopulusLog database contains 64,253 people, 3,250 locations, 11,235 organizations, 166,523 affiliations, and 11,242,354 pair-wise people relationships.

This paper is organized as follows. Section 2 describes data collection through crawling on the web. In Section 3, we describe PopulusLog's data extraction techniques along with its confidence scoring mechanism. Section 4 elaborates on the data mining tasks, such as home page identification, most informative web document set detection, locating similar people, and virtual social network computation. In Section 5, we comparatively discuss the related work, and Section 6 concludes with perspectives on future work.

II. DATA COLLECTION: CRAWLING THE WEB

A. Generating a Dictionary of Person Names

Initial stage of data collection starts with building a

* *These authors have contributed equally in this study.*

978-1-4244-5023-7/09/$25.00 ©2009 IEEE

ISCIS September 14-16, 2009
METU
Northern Cyprus Campus

dictionary for the names of people in a community. In this stage so far, we have collected names and email addresses of people from the CWRU (Case Western Reserve University) phone directory. In a general setting where PopulusLog is set to run on the whole web, we can obtain this information from named entity taggers, as well. To ensure privacy, we eliminate person names that do not exist on any public webpage. In addition, we do not publish any email addresses, and employ email identifiers for only authentication purposes.

B. Locating Web Pages

We have implemented a search engine crawler that queries search engines (i.e., Yahoo, Live Search, Google, Ask.com, and MetaCrawler), and stores the top k results of the query in our database. Since there is a daily limit of total number of queries that can be sent to a search engine, our crawler automatically switches between search engines on server error. Using our search engine crawler, we have collected links to web pages that contain a person name directly or in a varied forms (e.g., John Doe becomes J. Doe, or Doe, John etc.), and downloaded contents of those web pages. PopulusLog relies on MetaCrawler to search, combine, and rank results from different search engines.

III. DATA EXTRACTION

A. Extracting Entities

At this stage, we now have a set of documents ranked (with respect to the results of search engine queries) and associated with people. In the scope of this project, we are interested in person, location, and organization entities mentioned on web pages. We extract associated entities of people from web page contents. We have utilized two different named entity taggers, namely GATE [25] and OpenNLP [26], to locate entities in documents. These two taggers have different advantages (i.e., different accuracy and precision); hence we utilize both taggers simultaneously.

We have observed that essential information about people usually have no paragraph structure, and is distributed over the page in the form of item lists (e.g., CVs, publication lists). Thus, we use statistical measures to find such associations, instead of natural language processing [27][28][29].

Our statistical analysis consists of two parts; (i) distinguishing useful tags, and (ii) finding relationships between entities.

B. Scoring Extracted Entities

After extracting all entities from documents, we define rules to distinguish correct entities from incorrectly tagged ones.

Def'n *(Correctly Tagged Entity)*: Let M(E) be the number of documents that contain entity E (with or without being tagged) and $DE^T(G, E)$ be the number of documents that contain entity E tagged as the entity type T (i.e., location, organization) by tagger G (i.e., OPENNLP, GATE). We classify entities as "correctly tagged" if the following two rules hold:

- $DE^T(G_{GATE}, E) \geq F_1 \wedge DE^T(G_{OPENNLP}, E) \geq F_2 \wedge M(E) \geq F_3$
- $DE^T(G_{GATE}, E)/ M(E) \geq R_1 \wedge DE^T(G_{OPENNLP}, E)/ M(E) \geq R_2$

F_1, F_2, and F_3 are frequency thresholds. And, we have empirically pick F_1=10, F_2=20, and F_3=30. R_1 and R_2 are ratio thresholds to simulate the probability of a word being an entity of target type. We empirically set $R_1 = 0.15$ and $R_2 = 0.25$. We use thresholds F_1, F_2, and F_3 in order to eliminate falsely tagged character sequences that are derived from binary content.

Next, we associate entities (person, organization, location) with person entities. Associated entities are extracted from the web pages that include person names, and are the results of search engine queries. First, we compute an importance score for each person - web document (i.e., a web page) pair which represents how much information that the web document provides about the person.

We employ search engine ranking information to compute importance scores of web pages for a given person.

Def'n *(Importance Score)*: Let SER (D, P) be the rank of a document D (i.e., SER (D, P) is a positive integer equal to or greater than 1) in the search engine results of person P. Let SERC (P) be the number of all search engine results of person P, and NP (D) be the number of people that the document D appears in the search engine results for these people. Document-person importance score DPI (D, P) for document D and person P is defined as follows.

$$DPI(D, P) = SERC(P) / [NP(D) * SER(D, P)].$$

The motivation for constructing the above formula can be explained as follows. Importance of document D for person P, DPI (D, P), is inversely proportional to SER (D, P) since pages with higher search engine ranks (i.e., lower SER(D, P) means higher rank) are more likely to be informative for a person. A web page that contains a big list of people is considered not as important as another page that contains a single person's name. Therefore DPI (D, P) is also inversely proportional to NP (D). The same web page ranked as best within the query results for two different people is more important for the person who is mentioned by more web pages, due to the competition with more web pages for the top rank. Hence, SERC (P) is proportional to DPI (D, P).

Then, we normalize the importance score by dividing DPI (D, P) by the maximum score per person:

$$DPIN(D, P) = DPI(D, P) / Max_{D \in PD(P)}(DPI(D, P)),$$

where PD (P) is the search engine results of person P.

C. Defining and Scoring Relationships

We consider two people as related only when their names are mentioned together in at least one web page. We consider the following criteria while measuring the strength of the relationship between two people. First, we consider relationships between people as asymmetric. For example, when we find a blog where a person P posts his/her opinion about Tom Cruise's latest movie, we can say that the person P knows Tom Cruise, but it may not be true the other way around. Therefore, we use the search engine ranks of pages to create a directed association between person entities. Second, web pages are not equally important for people. We cannot decide about the strength of relationships by looking at a single web page, e.g., the page with the highest rank. All shared web pages of the person pair (i.e., web pages that mention the names of both people in the pair) should be considered. Thus, we employ (1) the total number of web pages that mention both people's names in order to measure

how frequent the two people are mentioned, and (2) the rankings of the pages that mention both people's names among the rankings of all pages that mention only one of the two people's names in order to measure the asymmetric association between the two people. For example, (1) the number of hits for the query "Tom Cruise" AND "Katie Holmes" is very high because these two people are related; and (2) the rank of the web page mentioning "Katie Holmes" is higher than the rank of the web page mentioning "Will Ferrell" among the web pages that mention "Tom Cruise"; thus we conclude that "Katie Holmes" is more related to "Tom Cruise" than "Will Ferrell" is.

We measure the strength of the relationship between two people by incorporating the factors explained above, namely (1) the highest rank of the document that mentions the names of both P_1 and P_2 among the documents that mention P_1, and (2) the total number of documents that mention the names of both P_1 and P_2:

$$PP(P_1, P_2) = \text{Max}_{D \in SD(P1, P2)} DPI\,N(D, P_1)) * |SD(P_1, P_2)|$$

where PP (P_1, P_2) is the strength of association of P_2 to P_1.

Next, we generalize this association strength measure to other entity types (i.e., organization and location) as well.

Def'n *(Relationship Strength):* Let $DEC^T(D, E)$ be the number of occurrences of entity E tagged as type T in document D. Combining importance and entity frequency, we compute strength $PE^T(P, E)$ of the relationship between person P and entity E of type T (T≠person) as follows:

$$PE^T(P, E) = \text{Max}_{D \in PD(P)} [DPIN(D, P) * DEC^T(D, E)]$$

where document D is an evidence for the relationship between P and E. Finally, we process all documents, and store the relationship between a person P and an entity E if the relationship strength is greater than a predefined threshold.

IV. MINING EXTRACTED DATA

A. Home Page Location

It is crucial to extract data about a person from the web page(s) that focuses the most on the person. To this end, despite some exceptions, personal home pages usually provide the most focused and complete information about an individual. A simple heuristic to use for homepage location would be based on the premise that, when searched by a person's full name on a search engine, the first resulting page would be his/her personal home page. However, our experience with the current data stored in PopulusLog indicates that there are many cases in which this premise does not hold. Moreover, even when a person has no personal home page, *the most informative* pages should be located and utilized as information sources. To this end, our approach employs supervised learning methods, namely, classification and rule-based techniques.

We have developed three different classifiers, namely, SVM-based classifier [23], Frequent Itemset-based classifier [24], and Rule-Based Classifier. Each of these classifiers is applied on the candidate web pages. Then, an overall confidence score is computed to decide the degree of *informativeness* of a web page. We define the informativeness of a web page as the probability that the web page is a home page of a person. Hence, a higher home page probability is interpreted as a high degree of informativeness for a given web page.

The success of supervised learning methods profoundly depends on the quality of the training data used to construct the internal models of the classifiers. Locating personal home pages manually even for training data may require significant amounts of time spent on searching for and collecting sufficient numbers of pages for training. Hence, to obtain the training data, we have used the pages listed in the "Personal Home Pages" directory in the DMOZ Open Directory Project [8]. As for negative (i.e., false) training data, we have collected random web pages as negative homepage examples.

For the final decision on whether a given a web page is a personal home page or not, we merge the assessments from both classification and rule-based methods in a weighted manner. In order to assess the individual reliability of the methods, first, each method is employed individually, and its performance is evaluated through k-fold cross validation [5]. Depending on the performances of individual methods, they are assigned relative weights for their contributions to the final assessment. Next, we describe the classifiers we have employed.

1) *Frequent Item Set-based Classifier:*

Frequent itemset mining was initially introduced to mine market basket data transactions in large databases [30]. Each document can also be considered as a bag of words, and the same technique can be applied to locate frequent word sets associated with home pages. The same frequent word sets may well be frequent also in non-home page documents. Therefore, we have also computed frequent word sets for negative training data. Each frequent word set is associated with a positive score and a negative score according to their frequency in both positive and negative training samples. As the minimum frequency threshold, we only retain the frequent word sets that appear in at least 20% of the positive and/or negative training data. We have also tried lower thresholds, but they do not improve the classification accuracy.

Then, the classification task boils down to checking each document if it contains a frequent word set. Accordingly, each frequent word set contributes to the score that whether a page is a home page or not in terms of the frequency values associated with each frequent word set. In cross validation experiments on training data, this classifier provides 61% accuracy. For the frequent itemset mining, we employ the data mining software package, IlliMine [6], which has the implementation for various data mining tasks including frequent itemset mining.

2) *Support Vector Machine-based Classifier:*

Support Vector Machines (SVM) [23] are widely used classification tools in data mining literature. SVM operates on high dimensional data, and attempts to locate the best hyper-plane that separates positive and negative data points in the most accurate way. In order to build an SVM-based classifier for home-page classification, one needs to find a feature vector representation of web documents. The most intuitive features for textual documents are words included in a document. TF/IDF [31] vectors are the most common ways to represent documents. We initially took this approach, but the

accuracy of the classification was not very promising as only 51% of the documents were correctly labeled as positive and negative.

Then, we have attempted to use a different feature set to represent documents. In this model, we have represented each document as a vector of frequent word sets that are computed previously for the frequent itemset-based classifier. Then, with this model, we have re-trained the SVM classifier, and obtained 62% accuracy. SVMs can be tuned by choosing a different kernel and/or adjusting various parameters associated with its internal classification model. However, it is quite counter-intuitive to reason about the optimal values of parameters due to the complexity of SVMs. Therefore, we have run an exhaustive search on parameter space to find the best parameter set for the existing training data. For this search we have utilized a script (GRID [7]) written in Python. It took 30 hours for this program to return optimal set of parameters. With the use of returned set of parameters, the cross-validation accuracy with frequent word sets as features reaches to 84%. We have also applied the same parameter tuning process for our previous unsuccessful attempt which uses TF/IDF vectors. However, using optimal set of parameters did not improve the accuracy of that model significantly, and provided 1% increase (52%) in comparison to its initial accuracy, 51%. For this classifier, we have used a software library [7] from National Taiwan University.

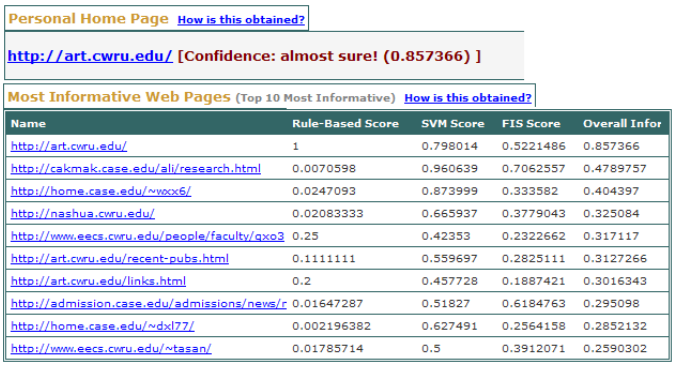


Figure 4.1: User home page, and most informative web pages for a person

3) Rule-based Identification of Personal Home Pages:

We have observed that personal home pages have common structures which can be formalized as rules signaling the personal nature of a web page, e.g., "Page Title Contains the text *Homepage of [Person A]*", "First Half of the Body Text Contains the text *Welcome to [Person A]'s web site*", etc.

We have developed a general rule-based classifier that can be extended with any rules describing a home page. Presently, as an instance of the rule-based classifier, we use a single rule that takes into consideration the rank of the page in the search result and the number of person entities that appear on the web page. This is also used during entity extraction and explained in detail in Section 3. Reliability of the rule-based score is assigned manually based on our confidence to the manually created rule that we used. According to our observations, the rule-based classifier assigns, most of the time, the highest scores to the most informative pages. Therefore, if a person has a home page in our database, the classifier finds it most the time. Hence, we assign the rule-based classifier a reliability score of 0.9. Problems occur if the user does not have a personal home page, but some other web pages that contain the name of the person. However, since we are using a combined overall scoring scheme, the other classifiers most of the time eliminate false positives caused by the rule-based classifier.

Figure 4.1 shows the home page of G. Ozsoyoglu, that is predicted successfully. We also display our confidence with scores as well as some English comments like "probably correct" about our estimations. Next, the figure shows the top 10 most informative pages located for the same person.

B. *Locating Similar People*

People may have similarities in terms of the entities that they are found to be related. PopulusLog employs three different similarity schemes, and combines outcomes of each similarity computation to compute an overall similarity score between people. The similarity is computed based on the number of shared locations, affiliations, and the web documents that two given persons appear. We use Jaccard measure [32] to compute each similarity, and then take the average of each similarity result. For instance, location similarity is the ratio of shared locations to the set of all locations associated with either of the two people.

$$Sim_{Loc}(Person1, Person2) = \frac{Person1_{Loc} \cap Person2_{Loc}}{Person1_{Loc} \cup Person2_{Loc}}$$

Organization and web document similarity Sim_{Org} is also defined in a similar way.

$$Sim_{org}(Person1, Person2) = \frac{Person1_{org} \cap Person2_{org}}{Person1_{org} \cup Person2_{org}}$$

$$Sim_{doc}(Person1, Person2) = \frac{Person1_{doc} \cap Person2_{doc}}{Person1_{doc} \cup Person2_{doc}}$$

Then, final information similarity is the sum of location and organization similarity. Final information similarity score can be extended as new entities are added into the system.

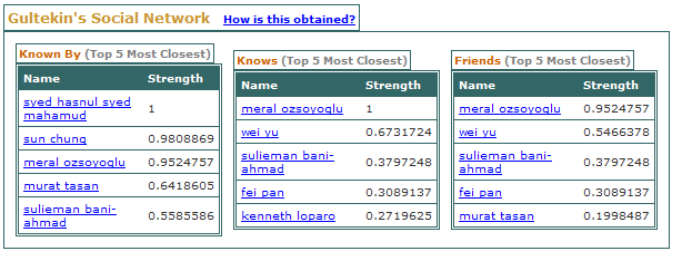
$$Sim_{overall}(Person1, Person2) = Avg(Sim_{loc} + Sim_{org} + Sim_{doc})$$

C. *Virtual Social Networks*

According to extracted person-person relations, we compute an impact factor for each person. The relationships between people are directed and weighted which shows the strength of the relationship between two people. In addition, we employ the PageRank algorithm [31] to compute the impact factors for individuals. A novel extension to the PageRank in PopulusLog is the assignment of weights to the relationships, and consideration of these relationship weights during impact factor computation.

Furthermore, we also provide a section in the person detail page to give an overview of social network of a person. This section includes three main relationship types, namely, people who know the person well (*known by*), people who are well known by the person (*knows*), and friends of the person. The first two relationship "knows" and "known by" are directly obtained from the person-person relationships extracted from web pages by our crawler. However, the "friends" relationship is computed and inferred implicitly from the existing relationship. In real life, if two people are friends, they know

each other very well. In addition, the relationship is reciprocal, that is, both people know each other at similar strengths. In this sense, the friendship relation is different than the "being a fan of somebody" in that "fan of" relation is most of the time unidirectional where a popular person is overly well known by many others although the popular person, most of the time, is not aware of the majority the people who know him very well. Based on this intuition, we select top-k people as friends of a person who knows them very well, they know him/her well, and the strength of the relation between them is similar; that is, there is no huge difference in different directions of a relationship between two friends. In order to keep the presentation concise, and eliminate the false positives, we only display top-k (k=5, presetly) people for each social network relationship. The figure below shows a screenshot for the social network of G. Ozsoyoglu (Figure 4.2).


Gultekin's Social Network How is this obtained?

Known By (Top 5 Most Closest)

Name	Strength
syed hasnul syed mahamud	1
sun chung	0.9808869
meral ozsoyoglu	0.9524757
murat tasan	0.6418605
sulieman bani-ahmad	0.5585586

Knows (Top 5 Most Closest)

Name	Strength
meral ozsoyoglu	1
wei yu	0.6731724
sulieman bani-ahmad	0.3797248
fei pan	0.3089137
kenneth loparo	0.2719625

Friends (Top 5 Most Closest)

Name	Strength
meral ozsoyoglu	0.9524757
wei yu	0.5466378
sulieman bani-ahmad	0.3797248
fei pan	0.3089137
murat tasan	0.1998487

Figure 4.2: Tabular social network presentation in PopulusLog

V. Related Work

In correlation with increasing numbers of Internet users, social networking web sites have been gaining a lot of interest. Through sites like Facebook [18], LinkedIn [19], MySpace [20], Orkut [21] and many others, it is possible to access detailed information about people. Personal information in these social networking sites is being created directly by the users themselves. On the other hand, our work proposed here targets information posted on all other web sites, rather than a single web profile of users. For example, in a particular community, such as within a university, or a company Intranet, personal homepages, internal project pages, wiki and forums provide rich information about this community. Hence, we compare our work with automated information retrieval studies that collect and aggregate information from public and semi-public (i.e., within Intranets) web pages.

First attempt to automatically create social networks is reported by Mika [12] as the *Flink* project. Flink system employs search engine co-occurrence frequencies in order to discover relevancy between entities, as described in [11]. However, Flink only focuses on RDF documents, and therefore requires information about people posted in structured format. In our work, we are interested in any text document, and in extracting entities from the text in an automated manner.

A following project for social network extraction is the *POLYPHONET* system developed by Matsuo *et al.* [13]. Matsuo *et al*. also employs search engine hit counts to measure whether two person names are related to each other. Furthermore, Matsuo *et al*. applies classification methods on lines of text where two person names are mentioned, in order to classify the relationship between people. On the other hand, although POLYPHONET finds frequent keywords located in the same text with person names, the system is not able to distinguish whether those keywords correspond to real world entities such as affiliations, occupations, or locations. In our work, we utilize state-of-the-art named entity tagging methodologies to discover entities (such as person, location, and organization names), and the relationships between those entities.

Cimple framework (that also houses DBLife) introduced in [14] differs from our work in the way of locating and expanding its database. DeRose *et al.* [14] starts with an expert-provided "high quality" data source list, and a dictionary of person names. Cimple's methodologies are source-aware; thus it has to know whether a page is a personal homepage, conference page, or an organization homepage. Then Cimple automatically discovers relationships between people (e.g., co-authorship), people and conferences (e.g., speaker or committee) and so on. Cimple is a highly accurate framework for building domain specific portals where activities of a list of known people are of interest. In contrast, PopulusLog aims to collect information about a very large group of people; hence it is not possible to build a dictionary of organization or affiliation names that the people of interest may have a relationship with.

A comparison of different frameworks for social information collection is displayed in Table 1. Missing features in our PopulusLog system such as classification of entity associations, and person name disambiguation will be discussed in the next section.

	PopulusLog	***Flink***	***POLYPHONET***	***Cimple***
Automated Homepage Identification	+	-	-	*Manual*
Discovering Associations between Named Entities	+	-	-	*Manual*
Classified Association Types	-	-	+	+
Impact Scores and Entity Ranking	+	-	-	-
Computing Similarity between Entities	+	-	-	-
Disambiguation of Person Names	*Manual*	-	+	+

Table 1: A Summary of Related Work Comparison

In addition to the projects above, recently a number of research prototypes and commercial products including EntityCube [15], Correlator [16], and Evri [17] have started to appear towards the direction of entity search, replacing traditional keyword search.

VI. Discussion and Future Work

In this paper, we have introduced the PopulusLog framework for collecting information about people in a target community. Using the PopulusLog framework, we have built personal profiles of students and faculty affiliated with CWRU. A framework like PopulusLog can be utilized for many applications. For example, automatically created personal profiles might be accessed by third parties to figure

out a person's role in a community. (1) Readers can find out about paper/book authors, (2) recruiters can confirm a job candidate's activities in the community that he/she claim to be a part of, and (3) people who are sensitive about their privacy can easily locate the information posted about them, and prevent any critical information leak. On the other hand, PopulusLog is still under development, and we have a number of future projects in mind to improve efficiency and accuracy of PopulusLog.

1. *Person name disambiguation.* We require a person name to be mentioned in a web page in order to be able to extract more information from that page. In the current design, when two people have the same name, we combine the information about both people on their profiles. A possible disambiguation methodology to be implemented is based on the association of entities to person names. For example, we expect two different people to be associated with different sub-communities, affiliations, and locations.

2. *Authenticity.* In the current PopulusLog design, we trust all the information found on the web pages. An improvement to this design is to incorporate actual search engine ranking scores (e.g., PageRank) for assessing authenticity of the information.

3. *Authority.* PopulusLog collects information by utilizing a list of names and identifiers (e.g., email addresses) of people in a community. Then, we employ the community identifiers to allow users to login to PopulusLog and update the information about them. In the cases where person identifiers are not known, currently we do not create any personal profiles. On the other hand, we can still extract email addresses, student/employee ids when they are available on web pages, and utilize such ids for users to claim authority to update their profiles.

4. *Privacy.* We do not create a profile for a person who is not mentioned on public web sites. In addition, we eliminate from our target list of people, the people who have asked for additional privacy from CWRU via FERPA agreement. However, there may still be privacy issues with the information that we are collecting. In some cases, we have received emails from users who did not know the existence of some private information posted on a public website, until it appeared on PopulusLog. We want to automatically detect web pages where unwanted information (e.g., SSN) is posted, and eliminate such pages from our extraction queue as well as informing users about such sites.

5. *Scalability.* The performance bottleneck in the current PopulusLog system is the NLP tools that allow us to extract entities (e.g., person, organization, location) from web pages. We want to replace our current named-entity-taggers with parallel tagging algorithms so that we can employ Map-Reduce techniques to increase the scalability of our system, and provide more up-to-date information.

6. *Extensions.* In addition to the relationships between person, location, and organization entities, we would like to extend the capability of PopulusLog to extract, classify, and index relationships between people and other entity types such as language, event, date, activities and interests.

REFERENCES

[1] PopulusLogCase People Information Database, CASE Edition, http://nashua.case.edu/PopulusLogCase

[2] Viladimir Vapnik. Statistical Learning Theory. John Wiley, 1998.

[3] Soumen Chakrabarti. Mining the Web: Discovering Knowledge from Hypertext Data. Morgan-Kauffman, 2002.

[4] Salton, G., Automatic Text Processing, Addison-Wesley, 1989.

[5] Jiawei Han, Micheline Kamber. Data Mining: Concepts and Techniques. The Morgan Kaufmann, 2000.

[6] IlliMine Data Mining Library, available for download at http://illimine.cs.uiuc.edu

[7] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[8] DMOZ Open Directory Project, available at http://dmoz.org

[9] WebSPHINX: A Personal, Customizable Web Crawler available at http://www.cs.cmu.edu/~rcm/websphinx/

[10] GATE: General Architecture for Text Engineering, http://www.gate.ac.uk/

[11] Kautz H, Selman B, and Shah M. The Hidden Web. AI Magazine, 18(2):27-36, 1997.

[12] Mika P. Flink: Semantic web technology for the extraction and analysis of social networks. Journal of Web Semantics, 3(2), 2005.

[13] Matsuo Y, Mori J, Hamasaki H. POLYPHONET: an advanced social network extraction system from the web. WWW-06.

[14] DeRose P, Shen W, Chen F, Doan A, Ramakrishnan R. Building Structured Web Community Portals: A TopDown, Compositional, and Incremental Approach. VLDB '07.

[15] EntityCube at Microsoft Research, homepage available at, http://research.microsoft.com/en-us/projects/entitycube

[16] Correlator at Yahoo! Research, prototype available at, http://correlator.sandbox.yahoo.net

[17] Evri search engine, system available at, http://www.Evri.com

[18] Facebook social networking, system available at, http://www.Facebook.com

[19] LinkedIn professional social networking, system available at, http://www.LinkedIn.com

[20] MySpace social networking, system available at, http://www.MySpace.com

[21] Orkut social networking, system available at, http://www.Orkut.com

[22] Gorton I, Greenfield P, Szalay A, and Williams R. Data-Intensive Computing in the 21st Century. Computer 41, 4 (Apr. 2008), 30-32.

[23] Corinna Cortes and V. Vapnik, "Support-Vector Networks", Machine Learning, 20, 1995.

[24] Agrawal R, Imielinski T, and Swami A. Mining association rules between sets of items in large databases. In Proc. of SIGMOD'93.

[25] Cunningham H, Maynard D, Bontcheva K, Tablan V. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. ACL'02.

[26] OpneNLP natural language processing tools. Project webpage available at http://opennlp.sourceforge.net

[27] S. Vogel, "PESA: Phrase pair extraction as sentence splitting," in Proc. of the Machine Translation Summit X, Phuket, Thailand, 2005.

[28] Ponzetto SP and Strube M. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In Proceedings of the Human Language Technolgy Conference of the North American Chapter of the Association for Computational Linguistics, 2006.

[29] Ponzetto SP, and Strube M. Semantic role labeling for coreference resolution. In Companion Volume of the Proceedings of the 11th Meeting of the European Chapter of the Association for Computational Linguistics, 2006.

[30] Agrawal R and Srikant R. Fast algorithms for mining association rules. VLDB 1994.

[31] Salton, G. 1989. Automatic Text Processing. The Transformation, Analysis and Retrieval of Informotion by Computer. Addison-Wesley.

[32] Tan P, Steinbach M and Kumar V. Introduction to Data Mining (2005)

[33] Brin S and Page L. The anatomy of a large-scale hypertextual Web search engine. WWW 1998:107-117.