POLYPHONET: An Advanced Social Network Extraction System from the Web

Yutaka Matsuo
National Institute of Advanced Industrial Science and Technology
Sotokanda 1-18-13, Tokyo 101-0021
Japan
y.matsuo@aist.go.jp

Junichiro Mori
University of Tokyo
Hongo 7-3-1, Tokyo 113-8656
Japan
jmori@mi.ci.i.u-tokyo.ac.jp

Masahiro Hamasaki
National Institute of Advanced Industrial Science and Technology
Aomi 2-41-6, Tokyo 135-0064
Japan
hamasaki@ni.aist.go.jp

ABSTRACT

Social networks play important roles in the Semantic Web: knowledge management, information retrieval, ubiquitous computing, and so on. We propose a social network extraction system called *POLYPHONET*, which employs several advanced techniques to extract relations of persons, detect groups of persons, and obtain keywords for a person. Search engines, especially Google, are used to measure co-occurrence of information and obtain Web documents.

Several studies have used search engines to extract social networks from the Web, but our research advances the following points: First, we reduce the related methods into simple pseudocodes using Google so that we can build up integrated systems. Second, we develop several new algorithms for social networking mining such as those to classify relations into categories, to make extraction scalable, and to obtain and utilize person-to-word relations. Third, every module is implemented in POLYPHONET, which has been used at four academic conferences, each with more than 500 participants. We overview that system. Finally, a novel architecture called *Super Social Network Mining* is proposed; it utilizes simple modules using Google and is characterized by scalability and *Relate-Identify processes*: Identification of each entity and extraction of relations are repeated to obtain a more precise social network.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information storage and retrieval

General Terms

Algorithms

Keywords

social network, search engine, Web mining

1. INTRODUCTION

Social networks play important roles in our daily lives. People conduct communications and share information through social relations with others such as friends, family, colleagues, collaborators, and business partners. Our lives are profoundly influenced by social networks without our knowledge of the implications. Potential applications of social networks in information systems are presented in [43]: Examples include viral marketing through social networks (also see [24]) and e-mail filtering based on social networks.

Social networking services (SNSs) have been given much attention on the Web recently. As a kind of online application, SNSs are useful to register personal information including a user's friends and acquaintances on these systems; the systems promote information exchange such as sending messages and reading Weblogs. Friendster[1] and Orkut[2] are among the earliest and most successful SNSs. Increasingly, SNSs especially target focused communities such as music, medical, and business communities. In Japan, one of large SNSs has more than three million users, followed by more than 70 SNSs that have specific characteristics for niche communities. Information sharing on SNSs is a promising application of SNSs [15, 35] because large amounts of information such as private photos, diaries and research notes are neither completely open nor closed: they can be shared loosely among a user's friends, colleagues and acquaintances. Several commercial services such as Imeem[3] and Yahoo! 360°[4] provide file sharing with elaborate access control.

In the context of the Semantic Web, social networks are crucial to realize a web of trust, which enables the estimation of information credibility and trustworthiness [16]. Because anyone can say anything on the Web, the web of trust helps humans and machines to discern which contents are credible, and to determine which information can be used reliably. Ontology construction is also related to a social network. For example, if numerous people share two concepts, the two concepts might be related [32, 33]. In addition, when mapping one ontology to another, persons between the two communities play an important role. Social networks enable us to detect such persons with high *betweenness*.

Several means exist to demarcate social networks. One approach is to make a user describe relations to others. In the social sciences, network questionnaire surveys are often performed to obtain social networks, e.g., asking "Please indicate which persons you would regard as your friend." Current SNSs realize such procedures online. However, the obtained relations are sometimes inconsistent; users do not name some of their friends merely because they are not in the SNS or perhaps the user has merely forgotten them. Some

WWW 2006, May 23–26, 2006, Edinburgh, Scotland.
ACM 1-59593-323-9/06/0005.

[1] http://www.friendster.com/
[2] http://www.orkut.com/
[3] http://www.imeem.com/
[4] http://360.yahoo.com//

name hundreds of friends, while others name only a few. Therefore, deliberate control of sampling and inquiry are necessary to obtain high-quality social networks on SNSs.

In contrast, automatic detection of relations is also possible from various sources of information such as e-mail archives, schedule data, and Web citation information [1, 44, 34]. Especially in some studies, social networks are extracted by measuring the co-occurrence of names on the Web. Pioneering work was done in that area by H. Kautz; the system is called Referral Web [21]. Several researchers have used that technique to extract social networks, as described in the next section.

This paper presents advanced algorithms for social network extraction from the Web. Our contributions are summarized as follows:

- Related studies are summarized and their main algorithms are described in brief pseudocodes. Surprisingly a few components that use Google consist of various algorithms.
- New aspects of social networks are investigated: classes of relations, scalability, and a person-word matrix.
- A social network mining system called *POLYPHONET* was developed and operated at the 17th, 18th and 19th Annual Conferences of the Japan Society of Artificial Intelligence (JSAI2003, JSAI2004, and JSAI2005) and at The International Conference on Ubiquitous Computing (UbiComp 2005) to promote participants' communication. More than 500 participants attended each conference; about 200 people actually used the system. We briefly overview that system.
- A novel architecture, called *Super Social Network Mining* is proposed. It is characterized by scalability and a Relate-Identify process.

Below, we take the JSAI cases as examples: a system is developed in Japanese language for JSAI conferences and in English language for the UbiComp conference. Differences of languages affect many details of algorithms. For that reason, we try to keep the algorithms as abstract as possible. We have various evaluations of algorithms of Japanese versions, but we have insufficient evaluations for the English version. Therefore, we show some evaluations in the Japanese version if necessary, in order to provide meaningful insights to readers.

This paper is organized as follows. The following section describes related studies and motivations. Section 3 addresses basic algorithms to obtain social networks from the Web. Advanced algorithms are described in Section 4 including evaluations. We briefly overview POLYPHONET in Section 5. We propose Super Social Network Mining architecture in Section 6 and conclude this paper.

2. RELATED WORK

In the mid-1990s, Kautz and Selman developed a social network extraction system from the Web, called *Referral Web* [21]. The system focuses on co-occurrence of names on Web pages using a search engine. It estimates the strength of relevance of two persons X and Y by putting a query "X and Y" to a search engine: If X and Y share a strong relation, we can find much evidence that might include their respective homepages, lists of co-authors in technical papers, citations of papers, and organizational charts. Interestingly, a path from a person to a person (e.g., from Henry Kautz to Marvin Minsky) is obtained automatically using the system. Later, with development of the WWW and Semantic Web technology, more information on our daily activities has become available online. Automatic extraction of social relations has much greater potential and demand now compared to when Referral Web is first developed.

Recently, P. Mika developed a system for extraction, aggregation and visualization of online social networks for a Semantic Web community, called Flink [32][5]. Social networks are obtained using analyses of Web pages, e-mail messages, and publications and self-created profiles (FOAF files). The Web mining component of Flink, similarly to that in Kautz's work, employs a co-occurrence analysis. Given a set of names as input, the component uses a search engine to obtain hit counts for individual names as well as the co-occurrence of those two names. The system targets the Semantic Web community. Therefore, the term "Semantic Web OR Ontology" is added to the query for disambiguation.

A. McCallum and his group [12, 3] present an end-to-end system that extracts a user's social network.That system identifies unique people in e-mail messages, finds their homepages, and fills the fields of a contact address book as well as the other person's name. Links are placed in the social network between the owner of the web page and persons discovered on that page. A newer version of the system targets co-occurrence information on the entire Web, integrated with name disambiguation probability models.

Other studies have used co-occurrence information: Harada et al. [19] develop a system to extract names and also person-to-person relations from the Web. Faloutsos et al. [14] obtain a social network of 15 million persons from 500 million Web pages using their co-occurrence within a window of 10 words. Knees et al. [22] classify artists into genres using co-occurrence of names and keywords of music in the top 50 pages retrieved by a search engine. Some particular social networks on the Web have been investigated in detail: L. Adamic has classified the social network at Stanford and MIT students, and has collected relations among students from Web link structure and text information [1]. Co-occurrence of terms in homepages can be a good indication to find communities, even obscure ones.

In the context of the Semantic Web, a study by Cimiano and his group is one of the most relevant works to ours. That system, Pattern-based ANnotation through Knowledge On the Web (PANKOW), assigns a named entity into several linguistic patterns that convey semantic meanings [9, 10]. Ontological relations among instances and concepts are identified by sending queries to a Google API based on a pattern library. Patterns that are matched most often on the Web indicate the meaning of the named entity, which subsequently enables automatic or semi-automatic annotation. The underlying concept of PANKOW, *self-annotating Web*, is that it uses globally available Web data and structures to annotate local resources semantically to bootstrap the Semantic Web.

Most of those studies use co-occurrence information provided by a search engine as a useful way to detect the proof of relations. Use of search engines to measure the relevance of two words is introduced in a book, *Google Hacks* [7], and is well known to the public. Co-occurrence information obtained through a search engine provides a large variety of new methods that had been only applicable to a limited corpus so far. This study seeks the potential of Web co-occurrence and describes novel approaches that can be accomplished surprisingly easily using a search engine.

We add some comments on the stream of research on Web graphs. Sometimes the link structure of Web pages is seen as a social network; a dense subgraph is considered as a community [23]. Numerous studies have examined these aspects of ranking Web pages (on a certain topic), such as PageRank and HITS, and identifying a set of Web pages that are densely connected. However, particular Web pages or sites do not necessarily correspond to an author or a group of authors. In our research, we attempt to obtain a social network in

[5] http://flink.semanticweb.org/. The system won a 1st prize at the Semantic Web Challenge in ISWC2004.

```
Algorithm 3.1: GOOGLECOOC($X, Y$)
  comment: Given person names X and Y, return the co-occurrence.
  $n_X \leftarrow GoogleHit(``X")$
  $n_Y \leftarrow GoogleHit(``Y")$
  $n_{X \wedge Y} \leftarrow GoogleHit(``X\ Y")$
  $r_{X,Y} \leftarrow CoocFunction(n_X, n_Y, n_{x \wedge y})$
  return $(r_{X,Y})$
```

Figure 1: Measure co-occurrence using *GoogleHit*.

```
Algorithm 3.2: GOOGLECOOCTOP($X, Y, k$)
  comment: Given person names X and Y, return the co-occurrence.
  $D_X \leftarrow GoogleTop(``X", k)$
  $D_Y \leftarrow GoogleTop(``Y", k)$
  $n_X \leftarrow NumEntity(D_X \cup D_Y, X)$
  $n_Y \leftarrow NumEntity(D_Y \cup D_Y, Y)$
  $n_{X \wedge Y} \leftarrow NumCooc(D_X \cup D_Y, X, Y)$
  $r_{X,Y} \leftarrow CoocFunction(n_X, n_Y, n_{X \wedge Y})$
  return $(r_{X,Y})$
```

Figure 2: Measure co-occurrence using *GoogleTop*.

which a node is a person and an edge is a relation, i.e., in Kautz's terms, a hidden Web. Recently, Weblogs have come to provide an intersection of the two perspectives. Each Weblog corresponds roughly to one author; it creates a social network both from a link structure perspective and a person-based network perspective.

3. SOCIAL NETWORK EXTRACTION

This section introduces the basic algorithm that uses a Web search engine to obtain a social network. Most related works use one of the algorithms in this section. We use JSAI cases as examples.

3.1 Basic algorithm

3.1.1 Nodes and Edges

A social network is extracted through two steps. First we set nodes, then we add edges. Some studies, including those addressing the Referral Web and McCallum's study, have employed expansion of the network, subsequently creating new nodes and finding new edges iteratively.

In our approach, similarly to that of Flink, nodes in a social network are given. In other words, a list of persons is given beforehand. We collect authors and co-authors who have presented works at past JSAI conferences; we posit them as nodes.

Next, edges between nodes are added using a search engine. For example, assume we are to measure the strength of relations between two names: Yutaka Matsuo and Peter Mika. We put a query *Yutaka Matsuo* AND *Peter Mika* to a search engine. Consequently, we obtain 44 hits[6]We obtain only 10 hits if we put another query *Yutaka Matsuo* AND *Lada Adamic*. *Peter Mika* itself generates 214 hits and *Lada Adamic* generates 324 hits. Therefore, the difference of hits by two names shows the bias of co-occurrence of the two names: *Yutaka Matsuo* is likely to appear in Web pages with *Peter Mika* than *Lada Adamic*. We can guess that Yutaka Matsuo has a stronger relationship with Peter Mika. Actually in this example, Yutaka Matsuo and Peter Mika participated together in several conferences; they also co-authored one short paper.

```
Algorithm 3.3: GETSOCIALNET($L$)
  comment: Given person list L, return a social network G.
  for each $X \in L$
    do set a node in $G$
  for each $X \in L$ and $Y \in L$
    do $r_{X,Y} \leftarrow GoogleCooc(X, Y)$
  for each $X \in L$ and $Y \in L$ where $r_{X,Y} > threshold$
    do set an edge in $G$
  return $(G)$
```

Figure 3: Extract social network using *GoogleCooc*.

```
Algorithm 3.4: EXPANDPERSON($X, k$)
  comment: Extract person names from the retrieved pages.
  $D \leftarrow GoogleTop(``X'', k)$
  $E \leftarrow ExtractEntities(D)$
  return $(E)$
```

Figure 4: Expand person names.

```
Algorithm 3.5: GOOGLECOOCCONTEXT($X, Y, W_X, W_Y$)
  comment: Given $X$, $Y$ and word(s) $W_X$, $W_Y$, return co-occurrence.
  $n_X \leftarrow GoogleHit(``X\ W_X")$
  $n_Y \leftarrow GoogleHit(``Y\ W_Y")$
  $n_{X \wedge Y} \leftarrow GoogleHit(``X\ Y\ W_X\ W_Y")$
  $r_{X,Y} \leftarrow CoocFunction(n_X, n_Y, n_{X \wedge Y})$
  return $(r_{X,Y})$
```

Figure 5: Measure co-occurrence with disambiguation.

That approach estimates the strength of their relation by co-occurrence of their two names. We add an edge between the two corresponding nodes if the strength of relations is greater than a certain threshold. Several indices can measure the co-occurrence [29]: matching coefficient, $n_{X \wedge Y}$; mutual information, $\log(n n_{X \wedge Y} / n_X n_Y)$; Dice coefficient, $(2 n_{X \wedge Y})/(n_X + n_Y)$; Jaccard coefficient, $(n_{X \wedge Y}/n_{X \vee Y})$; overlap coefficient, $(n_{X \wedge Y}/\min(n_X, n_Y))$; and cosine, $(n_{X \wedge Y}/\sqrt{n_X n_Y})$; where n_X and n_Y denote the respective hit counts of name X and Y, and $n_{X \wedge Y}$ and $n_{X \vee Y}$ denote the respective hit counts of "X AND Y" and "X OR Y".

Depending on the co-occurrence measure that is used, the resultant social network varies. Generally, if we use a matching coefficient, a person whose name appears on numerous Web pages will collect many edges. The network is likely to be decomposed into clusters if we use mutual information. The Jaccard coefficient is an appropriate measure for social networks: Referral web and Flink use this coefficient. In POLYPHONET, we use the overlap coefficient [30] because it fits our intuition well: For example, a student whose name co-occurs almost constantly with that of his supervisor strongly suggests an edge from him to the supervisor. A professor thereby collects edges from her students. We also verify that overlap coefficients perform well by investigating the probability of co-authorship [31].

Pseudocode that measures co-occurrence of two persons is shown in Fig. 1. In this paper, we define two functions in pseudocodes.

- *GoogleHit*: it returns the number of hits retrieved by a given query, and

[6] As of October, 2005 by Google search engine. The hit count is that obtained after the omission of similar pages by Google.

- *GoogleTop*: it returns k documents that are retrieved by a given query.

Those two functions play crucial roles in our research. *CoocFunction* is a co-occurrence index. In our case, it is defined as

$$f(n_X, n_Y, n_{X \wedge Y}) = \begin{cases} \dfrac{n_{X \wedge Y}}{min(n_X, n_Y)} & \text{if } n_X > k \text{ and } n_Y > k, \\ 0 & \text{otherwise} \end{cases}$$

We set $k = 30$ for the JSAI case. Alternatively, we can take some techniques for smoothing.

There is an alternative means to measure co-occurrence using a search engine, i.e., to use top retrieved documents, shown in Fig. 2. *NumEntity* returns the number of mentions in a given document set. *NumCooc* returns the number of co-occurrence of mentions in a given document set. Some of the related works employ this algorithm, in which we can use more tailored NLP methods. However, when the retrieved documents are much more numerous than k, we can process only a small fraction of the documents.

A social network is obtained using the algorithm in Fig. 3. For each pair of nodes where co-occurrence is greater than the threshold, an edge is invented. Eventually, a network *G=(V,E)* is obtained in which V is a set of nodes and E is a set of edges. Instead of using *GoogleCooc*, we can employ *GoogleCoocTop* in case that documents are not so large and more detailed processing is necessary. If we want to expand the network one node at a time, we can put in the algorithm a module shown in Fig. 4, in which *ExtractEntities* returns extracted person names from documents, and iterate the execution of the module.

Although various studies have applied co-occurrence by a search engine to extract a social network, most of them correspond to one of the algorithms described previously

3.2 Disambiguate a Person Name

More than one person might have the same name. Such namesakes cause problems when extracting a social network. To date, several studies have produced attempts at personal name disambiguation on the Web [3, 17, 26, 27]. In addition, the natural language community has specifically addressed name disambiguation as a class of word sense disambiguation [45, 28].

Bekkerman and McCallum uses probabilistic models for the Web appearance disambiguation problem [3]: the set of Web pages is split into clusters, then one cluster can be considered as containing only relevant pages: all other clusters are irrelevant. Li et al. proposes an algorithm for the problem of cross-document identification and tracing of names of different types [25]. They build a generative model of how names are sprinkled into documents.

These works identify a person from appearance in the text when a set of documents is given. However, to use a search engine for social network mining, a good keyphrase to identify a person is useful because it can be added to a query. For example, in the JSAI case, we use an affiliation (a name of organization one belongs to) together with a name. We make a query "X AND (A OR B OR ...)" instead of "X" where A and B are affiliations of X (including past affiliations and short name for the affiliation). Flink uses a phrase *Semantic Web OR Ontology* for that purpose.

In the UbiComp case, we develop a name-disambiguation module [4]. Its concept is this: for a person whose name is not common, such as *Yutaka Matsuo*, we need to add no words; for a person whose name is common, we should add a couple of words that best distinguish that person from others. In an extreme case, for a person whose name is very common such as *John Smith*, many words must be added. The module clusters Web pages that are retrieved by each name into several groups using text similarity. It then outputs characteristic keyphrases that are suitable for adding to a query. The pseudocode *GoogleCoocContext* to query a search engine with disambiguating keyphrases is shown in Fig. 5, which is slightly modified from *GoogleCooc*. We regard keyphrases to be added as a context of a person.

```
Algorithm 4.1: CLASSIFYRELATION($X, Y, k$)

comment: Given person names X and Y, return the class of relation.
$D_{X \wedge Y} \leftarrow GoogleTop(``X\ Y'', k)$
for each $d \in D_{X \wedge Y}$
  do $c_d \leftarrow Classifier(d, X, Y)$
$class \leftarrow determine\ on\ c_d \in D_{X \wedge Y}$
return $(class)$
```

Figure 6: Classify relation.

Table 2: Word groups (translated from Japanese).

Group	Words
A	publication, paper, presentation, activity, theme, award, authors, etc.
B	member, lab, group, laboratory, institute, team, etc.
C	project, committee
D	workshop, conference, seminar, meeting, sponsor, symposium, etc.
E	association, program, national, journal, session, etc.
F	professor, major, graduate student, lecturer, etc.

4. ADVANCED EXTRACTION

This section introduces novel algorithms that POLYPHONET uses for advanced social network extraction.

4.1 Class of Relation

Various interpersonal relations exist: friends, colleagues, families, teammates, and so on. RELATIONSHIP [13] defines more than 30 kinds of relationships we often have as a form of subproperty of the *knows* property in FOAF. For example, we can write "I am a collaborator of John (and I know him)" in our FOAF file. Various social networks are obtainable if we can identify such relationships. A person is central in the social network of a research community while not in the local community. Actually, such overlaps of communities exist often and have been investigated in social network analyses [46]. It also provides interesting research topics recently in the context of complex networks [40].

Through POLYPHONET, we target the relations in a researcher community. Among them, four kinds of relations are picked up because of the ease at identifying them and their importance in reality.

- Co-author: co-authors of a technical paper
- Lab: members of the same laboratory or research institute
- Proj: members of the same project or committee
- Conf: participants in the same conference or workshop

Each edge might have multiple labels. For example, X and Y have both "Co-author" and "Lab." relations.

We first fetch the top five pages retrieved by the *X AND Y* query, i.e., using *GoogleTop("X Y",5)*. Then we extract features from the content of each page, as shown in Table 1. Attributes NumCo,

Table 1: Attributes and possible values.

Attribute		Values
NumCo	Number of co-occurrences of X and Y	zero, one, or more_than_one
SameLine	Whether names co-occur at least once in the same line	yes, or no
FreqX	Frequency of occurrence of X	zero, one, or more_than_two
FreqY	Frequency of occurrence of Y	zero, one, or more_than_two
GroTitle	Whether any of a word group (A-F) appears in the title	yes or no (for each group)
GroFFive	Whether any of a word group (A-F) appears in the first five lines	yes or no (for each group)

Table 3: Obtained rules.

Class	Rule
co-author	SameLine=yes
Lab	(NumCo = more_than_one & GroTitle(D)=no & GroFFive(A) = yes & GroFFive(E) = yes)
	or (FreqX = more_than_two & FreqY = more_than_two & GroFFive(A) = yes & GroFFive(D)=no) or ...
Proj	(SameLine=no & GroTitle(A)=no & GroFFive(F)=yes) or ...
Conf	(GroTitle(A)=no & GroFFive(B)=no & GroFFive(D)= yes)
	or (GroFFive(A)=no & GroFFive(D)=no & GroFFive(E)= yes) or ...

FreqX, and FreqY relate to the appearance of name X and Y. Attributes GroTitle and GroFFive characterize the contents of pages using word groups defined in Table 2. We produced word groups by selecting high tf-idf terms using a manually categorized data set.

Figure 6 shows the pseudocode to classify relations. The *Classifier* indicates any one classifier used in machine learning such as Naive Bayes, maximum entropy or support vector machine. In the JSAI case, we use C4.5 [41] as a classifier. Using more than 400 pages to which we manually assigned the correct labels, classification rules are obtained. Some of those obtained rules are shown in Table 3. For example, the rule for Co-author is simple: if two names co-occur in the same line, they are classified as co-authors. However, the Lab relationship is more complicated.

Table 4 shows error rates of five-fold cross validation. Although the error rate for Lab is high, others have about a 10% error rate or less. Precision and recall are measured using manual labeling of an additional 200 Web pages. The Co-author class yields high precision and recall even though its rule is simple. In contrast, the Lab class gives low recall, presumably because laboratory pages have greater variety.

Obtaining the class of relationship is reduced to a text categorization problem. A large amount of research pertains to text categorization. We can employ more advanced algorithms. For example, using unlabeled data also improves categorization [38]. Relationships depend on the target domain; therefore, we must define classes to be categorized depending on a domain.

Vastly numerous pages exist on the Web. For that reason, the *ClassifyRelation* module becomes inefficient when k is large. Top-ranked Web pages do not necessarily contain information that is related to the purpose. One approach to remedy that situation is to organize a query in a more sophisticated way. For example, if we seek whether X and Y has Lab relations, we can organize a query such as "X Y (publication OR paper OR presentation)" by consulting Tables 2 and 3. This algorithm is not implemented in POLYPHONET, but it works well in our other study for extraction of a social network of corporations [20]. In Question Answering systems, query formulation is quite a common technique.

Table 4: Error rates of edge labels, precision and recall.

class	error rate	precision	recall
Co-author	4.1%	91.8% (90/98)	97.8% (90/92)
Lab	25.7%	70.9% (73/103)	86.9% (73/84)
Proj	5.8%	74.4% (67/90)	91.8% (67/73)
Conf	11.2%	89.7% (87/97)	67.4% (87/129)

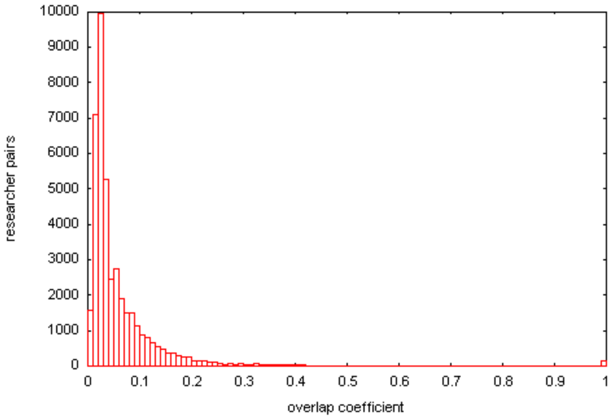


Figure 7: Number of pairs versus overlap coefficient.

4.2 Scalability

The number of queries to a search engine becomes a problem when we apply extraction of a social network to a large-scale community: a network with 1000 nodes requires 500,000 queries and grows with $O(n^2)$, where n is the number of persons. Considering that the Google API limits the number of queries to 1000 per day, the number is huge. Such a limitation might be reduced gradually with the development of technology, but the number of queries remains a great problem.

One solution might be found in the fact that social networks are often very sparse. For example, the network density of the JSAI2003 social network is 0.0196, which means that only 2% of possible edges actually exist. The distribution of the overlap coefficient is shown in Fig. 7. Most relations are less than 0.2, which is below the edge threshold. How can we reduce the number of queries while maintaining the extraction performance? Our idea is to filter out pairs of persons that seem to have no relation. That pseudocode is described in Fig. 8. This algorithm uses both good points of *GoogleCooc* and *GoogleCoocTop*. The latter can be executed in computationally low order (if k is a constant), but the former gives more precise co-occurrence information for the entire Web.

For 503 persons who participated in JSAI2003, ${}_{503}C_2 = 126253$

```
Algorithm 4.2: GETSOCIALNETSCALABLE($(L, k)$)

  comment: Given person list L, return a social network G.
  for each $X \in L$
    do set a node in $G$
  for each $X \in L$
    do $D \leftarrow GoogleTop(``X", k)$
       $E \leftarrow ExtractEntities(D)$
       for each $Y \in L \cap E$
         do $r_{X,Y} \leftarrow GoogleCooc(X, Y)$
  for each $X \in L\ and\ Y \in L$ where $r_{X,Y} > threshold$
    do set an edge in $G$
  return $(G)$
```

Figure 8: Extract social network in a scalable way.

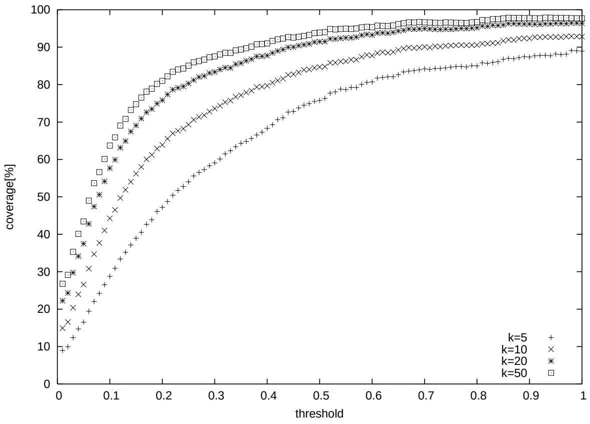


Figure 9: Coverage of *GetSocialNetScalable* for JSAI case.

queries are necessary if we use the *GetSocialNet* module. However, *GetSocialNetScalable* requires only 19,182 queries in case $k = 20$ empirically, which is about 15%. How correctly the algorithm filters out information is shown in Fig. 9. For example, in case $k = 20$, 90% or more of relations with an overlap coefficient 0.4 are detected correctly. It is readily apparent that as k increases, performance improves. (As an extreme case, we set $k = \infty$ and we achieve 100%.)

The computational complexity of this algorithm is $O(nm)$, where n is the number of persons and m is the average number of persons that remain candidates after filtering. Although m can be a function of n, it is bounded depending on k because a Web page contains a certain number of person names in the average case. Therefore, the number of queries is reduced from $O(n^2)$ to $O(n)$, which enables us to crawl a social network as large as $n = 7000$.[7]

4.3 Name and Word Co-occurrence

Person names co-occur along with many words on the Web. A particular researcher's name will co-occur with many words that are related to that person's major research topic. Below, we specifically address the co-occurrence of a name and words.

4.3.1 Keyword extraction

Keywords for a person, in other words personal metadata, are useful for information retrieval and recommendations on a social network. For example, if a system has information on a researcher's study topic, it is easy to find a person of a certain topic on a social network. PANKOW also provides such keyword extraction from a person's homepage [12].

```
Algorithm 4.3: EXTRACTKEYWORDS($(X, k_1, k_2)$)

  $D \leftarrow GoogleTop(X, k_1)$
  $words \leftarrow ExtractWords(D)$
  for each $W \in words$
    do $score_W \leftarrow GoogleCooc(X, W)$
  $K \leftarrow \{W | score_W\ is\ top\ k_2\}$
  return $(K)$
```

Figure 10: Extract keywords for a person.

```
Algorithm 4.4: CONTEXTSIM($(X, Y, W_L)$)

  comment: Given names X, Y and word list $W_L$, return the similarity.
  for each $W \in W_L$
    do $a_W \leftarrow GoogleCooc(X, W)$
       $b_W \leftarrow GoogleCooc(Y, W)$
  $s_{X,Y} \leftarrow$ similarity of two vectors $a = \{a_W\}$ and $b = \{b_W\}$
  return $(s_{X,Y})$
```

Figure 11: Measure context similarity of two persons.

In POLYPHONET, keyword extraction for researchers is implemented. A ready method to obtain keywords for a researcher is to search a person's homepage and extract words from the page. However, homepages do not always exist for each person. Moreover, a large amount of information about a person is not recorded in homepages, but is recorded in other resources such as conference programs, introductions in seminar Webpages, and profiles in journal papers. Therefore, POLYPHONET uses co-occurrence information to search the entire Web for a person's name.

We use co-occurrence of a person's name and a word (or a phrase) on the Web. The algorithm is shown in Fig. 10. Collecting documents retrieved by a person name, we obtain a set of words and phrases as candidates for keywords. We use Termex [37] for term extraction in Japanese as *ExtractWords*. Then, the co-occurrence of the person's name and a word / phrase is measured.

This algorithm is simple but effective. Figure 12 shows an example of keywords for Dan Brickley. He works with XML/RDF and metadata at W3C and ILRT; he created the FOAF vocabulary with Libby Miller. We can see that some important words, such as FOAF and Semantic Web, are extracted properly. Table 5 shows performance of the proposed algorithm based on a questionnaire. Both tf and tf-idf are baseline methods that extract keywords from D_X. In the tf- idf case, a corpus is produced by collecting 3981 pages for 567 researchers. For *ExtractKeywords*, we set $k_1 = 10$

Dan Brickley	Dan Connolly
Libby Miller	Jan Grant
FOAF	RDF Interest Group
Semantic Web	xmlns.com=foaf
Dave Beckett	RDF
RDFWeb	Eric Miller
ILRT	FOAF Explorer

Figure 12: Exemplary keywords for *Dan Brickley*.

[7] In case of the disaster mitigation research community in Japan.

Table 5: Precision and recall

Method	tf	tf-idf	*ExtractKeywords*
precision	0.13	0.18	0.60
recall	0.20	0.24	0.48

	W_1	W_2	W_3	$\dots$	W_m
X_1				$\dots$	
X_2				$\dots$	
X_3	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$\dots$				$\dots$	
X_n				$\dots$	

	X_1	X_2	$\dots$	X_M
X_1			$\dots$	
X_2			$\dots$	
X_3			$\dots$	
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
X_n			$\dots$	

Figure 13: Affiliation matrix and adjacent matrix.

and $k_2 = 20$ (as similarly as tf and tf-idf). We gave questionnaires to 10 researchers and defined the correct set of keywords carefully. (For details of the algorithm and its evaluation, see [36].) The tf outputs many common words; tf-idf outputs very rare words because of the diversity of Web document vocabularies. The proposed method is far superior to that of the baselines.

4.3.2 *Affiliation network*

Co-occurrence information of words and persons forms a matrix. Figure 13 shows a person-word co-occurrence matrix, which represents how likely a person's name co-occurs with words on the Web. In social network analysis literature, this matrix is called an *affiliation matrix* while a person-person matrix is called an *adjacent matrix* [46]. Figure 14 presents an example of a person-to-word matrix obtained in POLYPHONET. For example, the name of Mitsuru Ishizuka co-occurs often with words such as *agent* and *communication*. Koiti Hasida co-occurs often with *communication* and *cognition*. Our concept is that by measuring the similarity between two-word co-occurrence vectors (i.e., two rows of the matrix), we can calculate the similarity of the two people's contexts. In the researchers' cases, we can measure how mutually relevant the two researchers' research topics are: if two persons are researchers of very similar topics, the distribution of word co-occurrences will be similar.

Figure 11 describes the pseudocode for calculating the context similarity of two persons. We should prepare a word / phrase list W_L: a controlled vocabulary for the purpose, because rare words do not contribute much to the similarity calculation. In POLYPHONET, we obtain 188 words that appear frequently (excluding stop words) in titles of papers at JSAI conferences. Actually, we store the affiliation matrix for a list of persons and a list of words before calculating similarity to avoid inefficiency. Popular words such as *agent* and *communication* co-occur often with many person names. Therefore, statistical methods are effective: We first apply χ^2 statistics to the affiliation matrix and calculate cosine similarity [8].

One evaluation is shown in Fig. 15. Based on the similarity function, we plot the probability that the two persons will attend the same session at a JSAI conference. We compare several similarity calculations: chi^2 represents using the χ^2 and cosine similarity, the idf represents using idf weighting and cosine similarity, and hits represent using the hit count as weighting and cosine similarity. This session prediction task is very difficult and its precision and recall are low; the χ^2 performs best among the weighting methods.

A network based on an affiliation matrix is called *affiliation network* [46]. A relation between a pair of persons with similar interests or citations is sometimes called *intellectual link*. Even if no direct relation exists between the two, we can consider that they have common interests, implying a kind of intellectual relation, or potential social relation.

	agent	mining	communication	audio	cognition	...
Mitsuru Ishizuka	454	143	414	382	246	...
Koiti Hasida	412	156	1020	458	1150	...
Yutaka Matsuo	129	112	138	89	58	...
Nobuaki Minematsu	227	22	265	648	138	...
Yohei Asada	6	6	6	2	0	...
...	...	...	...	...	...	...

Figure 14: Example of a person-to-word co-occurrence matrix.

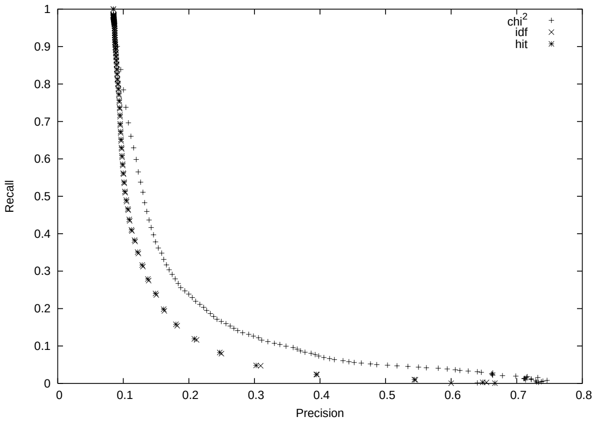


Figure 15: Precision and recall for session identification.

5. POLYPHONET

POLYPHONET is a coined term using *polyphony* + *network*. It is a Web-based system for an academic community to facilitate communication and mutual understanding based on a social network extracted from the Web. We implement every module mentioned above in POLYPHONET. The system has been used at JSAI annual conferences successively for three years and at UbiComp2005. Because of space limitations here, we briefly introduce the system. We encourage the reader to visit the website for UbiComp2005[8] and for JSAI2005[9].

A social network of participants is displayed in POLYPHONET to illustrate a community overview. Various types of retrieval are possible on the social network: researchers can be sought by name, affiliation, keyword, and research field; related researchers to a retrieved researcher are listed; and a search for the shortest path between two researchers can be made. Even more complicated retrievals are possible: e.g., a search for a researcher who is nearest to a user on the social network among researchers in a certain field. POLYPHONET is incorporated with a scheduling support system [18] and a location information display system [39] in the ubiquitous computing environment at the conference sites.

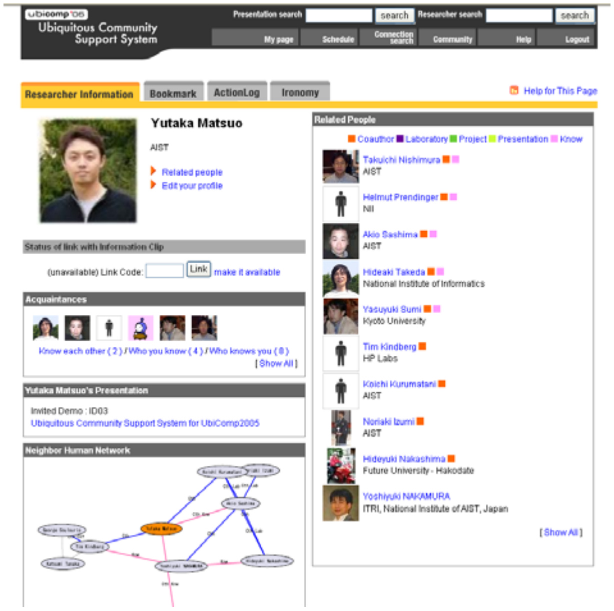
Figure 16 is a portal page that is tailored to an individual user, called *my page*. The user's presentations, bookmarks of the presentations, and registered acquaintances are shown along with the social network extracted from the Web. Figure 17 shows the obtained shortest path between two persons on a social network. Figure 18 is a screenshot that illustrates when three persons come to an

[8] http://www.ubicomp-support.org/ubicomp2005/.

[9] http://jsai-support-wg.org/polysuke2005/.

Table 6: Number of participants at conferences.

	JSAI03	JSAI04	JSAI05	UbiComp05
#participants	558	639	about 600	about 500
#users	276	257	217	308



ubicomp'05 Ubiquitous Community Support System

Presentation search [search] Researcher search [search]

My page | Schedule | Connection search | Community | Help | Logout

Researcher Information | Bookmark | ActionLog | Ironomy

Help for This Page

Yutaka Matsuo

AIST

- Related people
- Edit your profile

Status of link with Information Clip

(unavailable) Link Code: [Link] make it available

Acquaintances

Know each other (2) / Who you know (4) / Who knows you (0)

[Show All]

Yutaka Matsuo's Presentation

Invited Demo : ID03

Ubiquitous Community Support System for UbiComp2005

Neighbor Human Network

Related People

🟧 Coauthor 🟪 Laboratory 🟩 Project 🟩 Presentation 🟪 Know

Takuichi Nishimura 🟧 🟪
AIST

Helmut Prendinger 🟧 🟪
NII

Akio Sashima 🟧 🟪
AIST

Hideaki Takeda 🟧 🟪
National Institute of Informatics

Yasuyuki Sumi 🟧 🟪
Kyoto University

Tim Kindberg 🟧
HP Labs

Koichi Kurumatani 🟧
AIST

Noriaki Izumi 🟧
AIST

Hideyuki Nakashima 🟧
Future University - Hakodate

Yoshiyuki NAKAMURA
ITRI, National Institute of AIST, Japan

[Show All]

Figure 16: My page on POLYPHONET.

information kiosk and the social network including the three is displayed. More than 200 users used the system during each three-day conference, as shown in Table 6. Comments were almost entirely positive; they enjoyed using the system.

6. RELATE-IDENTIFY MODEL

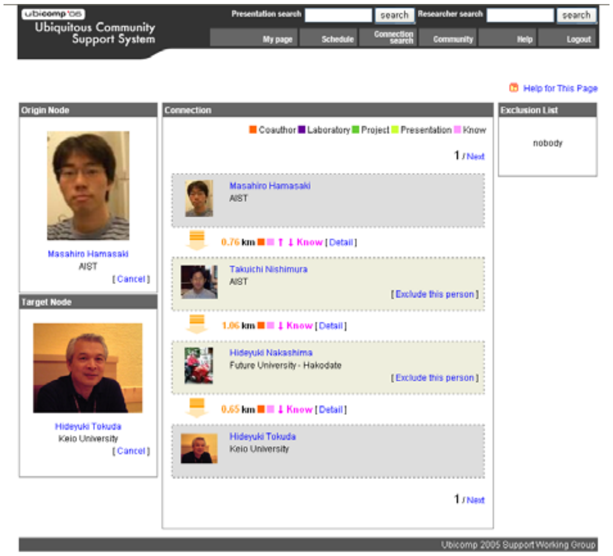
In this section, based on the studies of social network mining and lessons learned from POLYPHONET operation, we propose a novel architecture for social network extraction.

In the field of artificial intelligence, various forms of semantic representation have been speculated upon for decades, including first-order predicate logic, semantic networks, frames, and so on. Such representation enables us to describe relations among objects; it is useful for further use of the Web for integration of information and inference. On the other hand, studies of social network analyses in sociology provide us a means to capture the characteristics of a network as integration of relations. For example, the concept of centrality quantifies the degree to which a person is central to a social network. A measure of centrality, i.e., the degree to which a researcher is central to a research community, sometimes correlates to other measures of an individual, e.g., their number of publications. Social networks (and their individual relations) are defined properly in terms of a certain purpose if the correlation is high. Such feedback from an extracted network to individual relations is important when we target extraction of a large-scale social network from the Web.

Following that concept, we propose a new architecture to extract a social network from the Web, called *Super Social Network Mining*. The architecture has two characteristics:

Scalability We use very simple modules using a search engine to attain scalability.

Relate-Identify process We identify entities[10] and extract relations of entities. Then, based on the whole network structure and statistics, we improve the means to identify entities. That process is repeated iteratively; thereby, it is gradually improved.



ubicomp'05 Ubiquitous Community Support System

Presentation search [search] Researcher search [search]

My page | Schedule | Connection search | Community | Help | Logout

Help for This Page

Origin Node

Masahiro Hamasaki
AIST
[Cancel]

Target Node

Hideyuki Tokuda
Keio University
[Cancel]

Connection

🟧 Coauthor 🟪 Laboratory 🟩 Project 🟩 Presentation 🟪 Know

1 / Next

Masahiro Hamasaki
AIST

0.76 km 🟧 🟪 ↑ ↓ Know [Detail]

Takuichi Nishimura
AIST
[Exclude this person]

1.06 km 🟧 🟪 ↓ Know [Detail]

Hideyuki Nakashima
Future University - Hakodate
[Exclude this person]

0.65 km 🟧 🟪 ↓ Know [Detail]

Hideyuki Tokuda
Keio University

1 / Next

Exclusion List

nobody

Ubicomp 2005 Support Working Group

Figure 17: Shortest path from a person to a person on POLYPHONET.

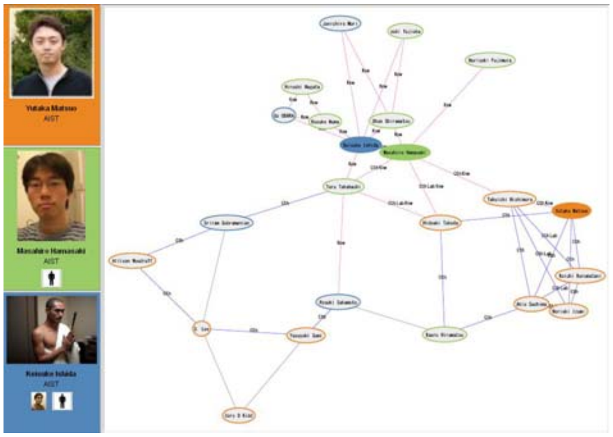


Figure 18: Social network among three persons on POLYPHONET.

To attain scalability, we allow two operations using a search engine: *GoogleTop* and *GoogleCooc*. These two are permissible operations even if the Web grows more. *GoogleTop* enables us to investigate a small set of samples of Web pages using text processing, whereas *GoogleCooc* provides statistics that pertain to the entire Web. We should note that as the Web grows, *GoogleTop* returns fewer and fewer Web pages relative to all retrieved documents, thereby rendering it less effective. A more effective means to sample documents from the Web is essential, as described in [2]. In contrast, *GoogleCooc* yields a more precise number if the Web grows because the low-frequency problem is improved. Therefore, a good combination of *GoogleCooc* and *GoogleTop* is necessary for Super Social Network Mining. For other kinds of operations by a search engine such as "get the number of documents where word X co-occurs with Y within the word distance of 10," whether they are permissible or not remains unclear in terms of scalability because the index size grows very rapidly. A search engine that is specially designed for NLP [6] will benefit our research greatly if it actually

[10] We use an *entity* as a broader term of a *person*.

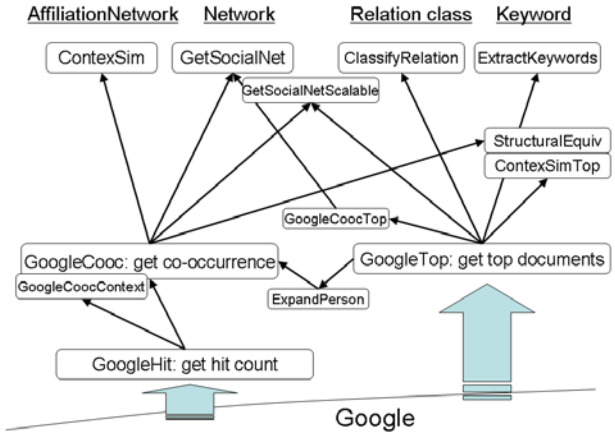


Figure 19: Overview of module dependency.

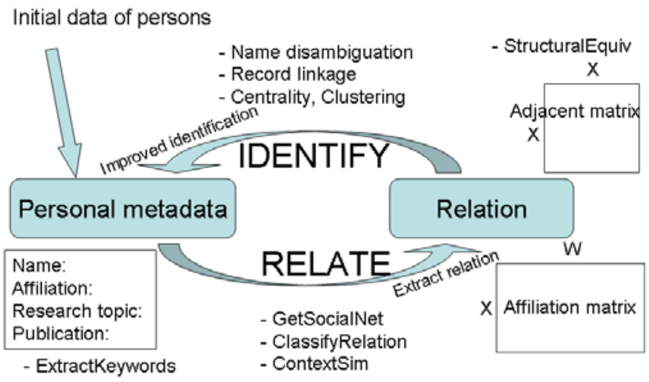


Figure 20: Relate-Identify process of *Super Social Network Mining*.

scales properly.

Figure 19 shows an overview of the module dependencies we described in this paper. *GoogleHit* and *GoogleTop* are remarkably versatile yet simple modules. We should note that two modules exist that we do not introduce in this paper: *ContextSimTop* and *StructuralEquiv*. The first, *ContextSimTop*, calculates the context similarity of two persons based on *GoogleTop*. That module is similar to the snippet similarity of two queries (or two short texts) introduced in [42]. The *StructuralEquiv* module calculates structural equivalence, which plays an important role in the Relate-Identify process.

Figure 20 depicts an overview of the Relate-Identify process. First, a list of names is given as the initial input. We apply the *ExtractKeyword* module to obtain some keywords that are useful for personal metadata. Then in the RELATE step, relations among persons are extracted using various modules including *GetSocialNet* and *ClassifyRelation*, which will eventually produce two kinds of matrices: an adjacent matrix and an affiliation matrix.

In the IDENTIFY step, information associated with overall relations is used to obtain an improved query for each person. Two possibilities to modify identification of an entity (or a person) exist: to decompose one entity into two or more, and to merge multiple entities into one. Decomposition of one entity is equivalent to name disambiguation, which is mentioned in the paper. Fundamentally, the *GoogleTop* module is used to obtain documents of a name, and then cluster the documents in some way. New keywords are obtained to identify the person more precisely.

Integration of multiple entities is known as a record linkage problem in database studies. In the context of social networks, examples include integrating a person with multiple names such as *James Hendler* and *Jim Hendler*, a person with different affiliations (as researchers often move institutes), and a person with multiple names in different languages. We propose the use of structural equivalence as a key to uncover entity linkage. Structural equivalence is the degree to which two individuals have the same relations with the same others [5]. The two names might refer to the same individual if the two entities have a very similar distribution of co-occurrence with others. Furthermore, we can use other information simultaneously: whether the two have similar keywords that are obtained by *ContextSim* module, and whether the two expressions of names share some proximity such as *Jim Hendler*, *James Hendler*, or *J. Hendler*.

Although the overall architecture is not implemented in POLYPHONET, we have partially implemented the system and the results appear promising. We can gradually obtain an improved query for each; simultaneously, the system has served to improve relations among individuals. We believe that this architecture works not only for social network extraction in the Japanese language, but also in other languages.

7. CONCLUSION

This paper describes a social network mining approach using the Web. Several studies have addressed similar approaches so far; we organize those methods into small pseudocodes. Several algorithms, which classify the relations using Google, make the extraction scalable, and obtain a person-to-word matrix, are novel as far as we know. We implemented every algorithm on POLYPHONET, which was put into service at JSAI conferences over three years and at the UbiComp conference. Finally, the Super Social Network Mining concept is proposed: it is characterized by its scalability and Relate-Identify process.

Merging the vast amount of information on the Web and producing higher-level information might contribute many knowledge-based systems in the future. Acquiring knowledge through Googling is a similar concept to ours [11]. We intend to apply our approach in the future to extract much structural knowledge aside from social networks.

8. ACKNOWLEDGEMENTS

This research was partially supported by NEDO (New Energy and Industrial Technology Development Organization), No. 04A11502a.

9. ADDITIONAL AUTHORS

Additional authors: Keisuke Ishida (National Institute of Advanced Industrial Science and Technology (AIST)), Takuichi Nishimura (AIST), Hideaki Takeda (NII), Koiti Hasida (AIST), Mitsuru Ishizuka (University of Tokyo)

10. REFERENCES

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[2] A. Anagnostopoulos, A. Z. Broder, and D. Carmel. Sampling search-engine results. In *Proc. WWW 2005*, pp. 245–256, 2005.

[3] R. Bekkerman and A. McCallum. Disambiguating web appearances of people in a social network. In *Proc. WWW 2005*, 2005.

[4] D. Bollegara, Y. Matsuo, and M. Ishizuka. Extracting key phrases to disambiguate personal names on the web. In *Proc. CICLing 2006*, 2006.

[5] R. S. Burt. *Structural Holes: The Social Structure of Competition*. Harvard University Press, Cambridge, MA, 1992.

[6] M. Cafarella and O. Etzioni. A search engine for natural language applications. In *Proc. WWW2005*, 2005.

[7] T. Calishain and R. Dornfest. *Google Hacks: 100 Industrial-Strength Tips & Tools*. O'Reilly, 2003.

[8] S. Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, 2002.

[9] P. Cimiano, S. Handschuh, and S. Staab. Towards the self-annotating web. In *Proc. WWW2004*, pp. 462–471, 2004.

[10] P. Cimiano, G. Ladwig, and S. Staab. Gimme´ the context: Context-driven automatic semantic annotation with cpankow. In *Proc. WWW 2005*, 2005.

[11] P. Cimiano and S. Staab. Learning by googling. *SIGKDD Explorations*, 6(2):24–33, 2004.

[12] A. Culotta, R. Bekkerman, and A. McCallum. Extracting social networks and contact information from email and the web. In *CEAS-1*, 2004.

[13] I. Davis and E. V. Jr. RELATIONSHIP: A vocabulary for describing relationships between people. http://vocab.org/relationship/.

[14] C. Faloutsos, K. S. McCurley, and A. Tomkins. Fast discovery of connection subgraphs. In *Proc. ACM SIGKDD 2004*, 2004.

[15] J. Goecks and E. D. Mynatt. Leveraging social networks for information sharing. In *Proc. ACM CSCW 2004*, pp. 328–331, 2004.

[16] J. Golbeck and J. Hendler. Accuracy of metrics for inferring trust and reputation in semantic web-based social networks. In *Proc. EKAW 2004*, 2004.

[17] R. Guha and A. Garg. Disambiguating entities in web search. TAP project, http://tap.stanford.edu/PeopleSearch.pdf.

[18] M. Hamasaki, H. Takeda, I. Ohmukai, and R. Ichise. Scheduling support system for academic conferences based on interpersonal networks. In *Proc. ACM Hypertext 2004*, 2004.

[19] M. Harada, Sh. Sato, and K. Kazama. Finding authoritative people from the web. In *Proc. Joint Conference on Digital Libraries (JCDL2004)*, 2004.

[20] Y. Jin, Y. Matsuo, and M. Ishizuka. Extracting inter-business relationship from world wide web. In *Workshop Notes, Web Community Structure and Network Analysis Workshop*, 2005.

[21] H. Kautz, B. Selman, and M. Shah. The hidden Web. *AI magazine*, 18(2):27–35, 1997.

[22] P. Knees, E. Pampalk, and G. Widmer. Artist classification with web-based data. In *5th International Conf. on Music Information Retrieval(ISMIR)*, 2004.

[23] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The web and social networks. *IEEE Computer*, 35(11):32–36, 2002.

[24] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing, 2005. http://www.hpl.hp.com/research/idl/papers/viral/viral.pdf.

[25] X. Li, P. Morie, and D. Roth. Semantic integration in text: From ambiguous names to identifiable entities. *AI Magazine Spring*, pp. 45–68, 2005.

[26] L. Lloyd, V. Bhagwan, D. Gruhl, and A. Tomkins. Disambiguation of references to individuals. Technical Report RJ10364(A0410-011), IBM Research, 2005.

[27] B. Malin. Unsupervised name disambiguation via social network similarity. In *Workshop Notes on Link Analysis, Counterterrorism, and Security*, 2005.

[28] G. S. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *Proc. CoNLL*, 2003.

[29] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*. The MIT Press, London, 2002.

[30] Y. Matsuo, H. Tomobe, K. Hasida, and M. Ishizuka. Finding social network for trust calculation. In *Proc. 16th European Conference on Artificial Intelligence (ECAI2004)*, pp. 510–514, 2004.

[31] Y. Matsuo, H. Tomobe, K. Hasida, and M. Ishizuka. Social network extraction from the web information. *Journal of the Japanese Society for Artificial Intelligence*, 20(1E):46–56, 2005. in Japanese.

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

[33] P. Mika. Ontologies are us: A unified model of social networks and semantics. In *Proc. ISWC2005*, 2005.

[34] T. Miki, S. Nomura, and T. Ishida. Semantic web link analysis to discover social relationship in academic communities. In *Proc. SAINT 2005*, 2005.

[35] J. Mori, M. Ishizuka, T. Sugiyama, and Y. Matsuo. Real-world oriented information sharing using social networks. In *Proc. ACM GROUP'05*, 2005.

[36] J. Mori, Y. Matsuo, and M. Ishizuka. Finding user semantics on the web using word co-occurrence information. In *Proc. Int'l. Workshop on Personalization on the Semantic Web (PersWeb05)*, 2005.

[37] H. Nakagawa, A. Maeda, and H. Kojima. Automatic term recognition system termextract. http://gensen.dl.itc.utokyo.ac.jp/gensenweb_eng.html.

[38] K. Nigram, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using *Machine Learning*, 39:103–134, 2000.

[39] T. Nishimura, Y. Nakamura, H. Itoh, and H. Nakamura. System design of event space information support utilizing CoBITs. In *Proc. IEEE ICDCS2004*, pp. 384–387, 2004.

[40] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814, 2005.

[41] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, California, 1993.

[42] M. Sahami and T. Heilman. A web-based kernel function for matching short text snippets. In *International Workshop on Learning in Web Search (LWS2005)*, pp. 2–9, 2005.

[43] S. Staab, P. Domingos, P. Mika, J. Golbeck, L. Ding, T. Finin, A. Joshi, A. Nowak, and R. Vallacher. Social networks applied. *IEEE Intelligent systems*, pp. 80–93, 2005.

[44] J. Tyler, D. Wilkinson, and B. Huberman. *Email as spectroscopy: automated discovery of community structure within organizations*, pp. 81–96. Kluwer, B.V., 2003.

[45] N. Wacholder, Y. Ravin, and M. Choi. Disambiguation of proper names in text. In *Proc. 5th Applied Natural Language Processing Conference*, pp. 202–208, 1997.

[46] S. Wasserman and K. Faust. *Social network analysis. Methods and Applications*. Cambridge University Press, Cambridge, 1994.