Title: POLYPHONET: An Advanced Social Network Extraction System from theWeb

Authors: Yutaka Matsuo, Junichiro Mori, Masahiro Hamasaki, Takuichi Nishimura, Hideaki Takeda, Koiti Hasida, Mitsuru Ishizuka

PII: S1570-8268(07)00034-0
DOI: doi:10.1016/j.websem.2007.09.002
Reference: WEBSEM 111

To appear in: *Web Semantics: Science, Services and Agents on the World Wide Web*

Received date: 24-5-2007
Revised date: 2-9-2007
Accepted date: 2-9-2007

Please cite this article as: Y. Matsuo, J. Mori, M. Hamasaki, T. Nishimura, H. Takeda, K. Hasida, M. Ishizuka, POLYPHONET: An Advanced Social Network Extraction System from theWeb, *Web Semantics: Science, Services and Agents on the World Wide Web* (2007), doi:10.1016/j.websem.2007.09.002

POLYPHONET: An Advanced Social Network Extraction System from the Web

Yutaka Matsuo [a] Junichiro Mori [b] Masahiro Hamasaki [a] Takuichi Nishimura [a] Hideaki Takeda [b]
Koiti Hasida [a] Mitsuru Ishizuka [b]

[a] *National Institute of Advanced Industrial Science and Technology*
[b] *University of Tokyo*

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, to detect groups of persons, and to 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 network 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 *Iterative 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.

Key words: social network, search engine, Web mining
PACS:

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. Social studies have been conducted from the 1930s, observing and modeling the network structure, its influence and dynamics, and information flow from tribal and village societies on to global corporate and industrial societies. Direct applications of social networks in information systems are presented in [1]: Examples include viral marketing through social networks (also see [2]) and e-mail filtering based on social networks.

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 [3]. Anyone can say anything on the Web. For that reason, the web of trust helps humans and machines to discern, reliably, which contents are credible, and to determine which information is useful. Ontology construction is also related to a social network. For example, if numerous people share two concepts, the two concepts might be related [4,5]. As P. Mika pointed out, social networks and semantics are merely sides of the same coin [5]. Ontologies are inseparable from the context of the community in which they are created and used. 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 friends." 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 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 [6–8]. 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 [9]. 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 the basic retrieval functions of a search engine 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–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 *Iterative Social Network Mining* is proposed. It is characterized by scalability and a Relate-Identify process.

Extracting a social network can be reduced into recognizing relations among entities. Relation extraction, representation of the relation (typically in RDF or OWL formats), and advanced usage of the multiple relations are a core of Semantic Web research. Although our approach is motivated by a network perspective, the core technique is the same, i.e., to recognize whether a relation exists between two entities, Therefore, our study can be applicable to ontology construction and advanced inference based on the extracted relations (e.g. [10]).

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 language affect many details of algorithms. For that reason, we try to keep the algorithms as abstract as possible. Various evaluations of algorithms of Japanese versions are available, but we have insufficient evaluations for the English version. We show some evaluations in the Japanese version as 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 Iterative Social Network Mining architecture in Section 6 and conclude this paper.

2. RELATED WORK

Although our research is based on the recent development of Web data and a search engine, a relevant work dates back to more than 10 years ago. In the mid-1990s, Kautz and Selman developed a social network extraction system from the Web, called *Referral Web* [9]. The system addresses 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 than 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 [4] [1] . 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 [11,12] 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. [13] developed a system to extract names and also person-to-person relations from the Web. Faloutsos et al. [14] obtained 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. [15] classified 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 [6]. Co-occurrence of terms in homepages

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

can be a good indication to find communities, even obscure ones.

The co-occurrence of terms is used to recognize synonyms and related terms in natural language processing (NLP) studies: As study by P. Turney [16] presents an unsupervised learning algorithm for recognizing synonyms by querying a web search engine. The task of recognizing synonyms is, given a target word and a set of alternative words, to choose the word that is most similar in meaning to the target word. The algorithm uses pointwise mutual information (PMI-IR) to measure the similarity of pairs of words. It is evaluated using 80 synonym test questions from the Test of English as a Foreign Language (TOEFL) and 50 from the English as a Second Language test (ESL). The algorithm obtains a score of 74%, contrasted to that of 64% by Latent Semantic Analysis (LSA). Another study by Terra and Clarke [17] describes a comparative investigation of co-occurrence frequency estimation on the performance of synonym tests. Based on that study, they conclude that PMI (with a certain window size) performs best on average.

A notable characteristic of such NLP studies compared to studies of social network mining is that they try to recognize the relation among general words, or classes, rather than a particular set of named-entities. They try to recognize synonyms, hypernyms-hyponyms, or coordinates (words with the same hypernym). However, the difference is minor at least from algorithmic perspective; if we apply one method to more concrete entities, we can extract a network of named entities, while if we apply the method to more abstract concepts, we can extract a network of concepts. Actually, we can see two such applications in Mika's studies [4,5], where he applies a similar algorithm to named entities (researcher names) and more generic concepts. We also apply a similar algorithm for social network mining among researchers (as we describe below), social network mining for companies and artists [18], and obtaining a word network (and derived clusters)[19].

In the context of the Semantic Web, a study by Cimiano and his group target a broader range of instances and concepts. That system, Pattern-based ANnotation through Knowledge On the Web (PANKOW), assigns a named entity into several linguistic patterns that convey semantic meanings [20,21]. 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 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* [22], 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. This study seeks the potential of Web co-occurrence and describes novel approaches that can be accomplished surprisingly easily using a search engine. In contrast to several related works, one strength of the study is that evaluation takes two forms in the paper: through comparison to a survey and through applications (by operation of POLYPHONET).

In terms of social networks, which are not necessarily relevant to co-occurrence, analysis of FOAF networks is a new research topic. To date, a couple of interesting studies have analyzed FOAF networks [23,4]. Aleman-Meza et al. proposed the integration of two social networks: "knows" from FOAF documents and "co-author" from the DBLP bibliography [24]. They integrate the two networks by weighting each relationship to determine the degree of Conflict of Interest among scientific researchers.

As one so-called "Web 2.0" services, social networking services (SNSs) have been given much attention on the Web recently. Actually, 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[2] and Orkut[3] are among the earliest and most successful SNSs, and MySpace[4] is currently the largest service. Increasingly, SNSs especially target focused communities such as music, medical, and business communities. In Japan, one large SNS 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 [25,26] 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[5] and Yahoo! 360°[6] provide file sharing with elaborate access control.

We also offer some comments on the stream of research related to Web graphs. Sometimes the link structure of Web pages is seen as a social network; a dense subgraph is considered as a community [27]. 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 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.

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

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 algorithm described in this section. We use JSAI cases as examples.

3.1. *Basic algorithm*

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 used 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 [7]. 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 more 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.

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 [28]: matching coefficient, $n_{X\wedge Y}$; mutual information, $\log(nn_{X\wedge Y} \,/\, n_X n_Y)$; Dice coefficient, $(2n_{X\wedge Y})/(n_X + n_Y)$; 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 [29] 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 should note that most co-occurrence measures described above are based on ratios which can change considerably if the number is low. Therefore, some absolute threshold for support is necessary in actual use. The combination of co-occurrence measures is also a solution to handle the noisy co-occurrence counts [18].

Figures 1-4 show the difference of the similarity measures. The Y-axis is the probability of co-authorship between a pair of JSAI researchers. The X-axis is the value of the measure. Although what we want to recognize is the strength of relations (and not simply the existence of co-authorship), it can be considered as a good proxy of relational strength in the researcher case to compare the effectiveness of similarity measures. Clear correlation is preferable.

It is readily apparent from Fig. 1 that a good correlation exists between the matching coefficient and the probability. Although the probability is not always near 1.0, the coefficient can be more than 100 because a person with many hits tends to produce large matching coefficients, simply because their name is likely to appear in many web pages. The Jaccard coefficient is better, as presented in Fig. 2; the probability is almost always 1.0 if the value of the coefficient is more than 0.1. For that reason, previous studies use the Jaccard coefficient rather

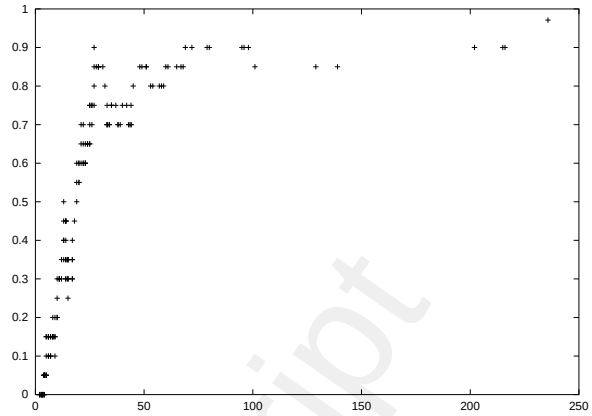

Fig. 1. Probability of co-authorship versus the matching coefficient.

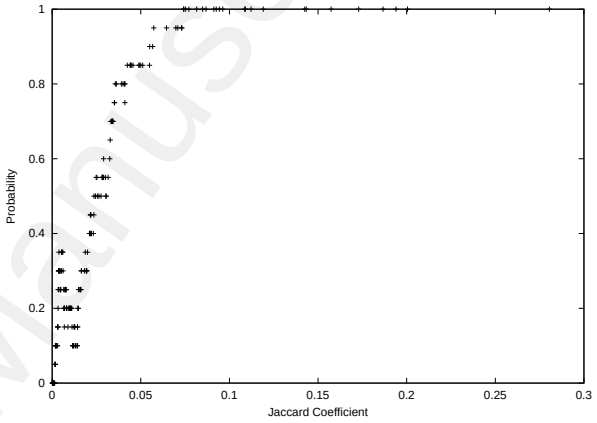

Fig. 2. Probability of co-authorship versus the Jaccard coefficient.

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

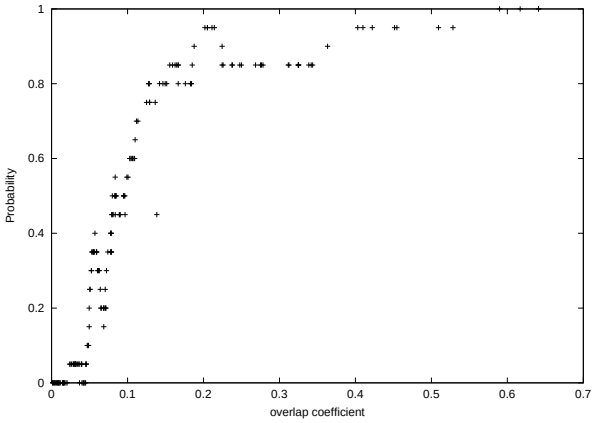

Fig. 3. Probability of co-authorship versus the overlap coefficient (without a threshold).

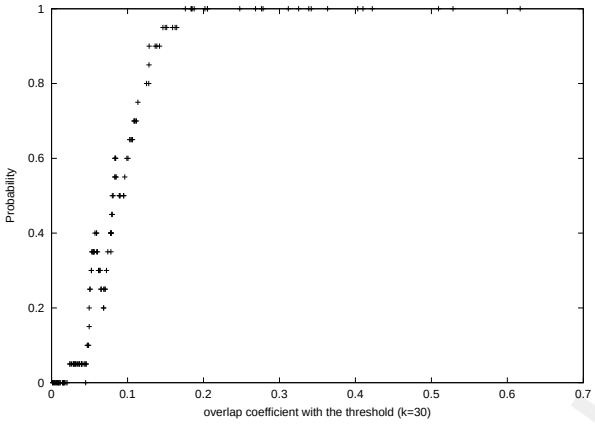

Fig. 4. Probability of co-authorship versus the overlap coefficient (without a threshold).

Algorithm 3.1: GOOGLECOOC(X, Y)

comment: Given person names X and Y, return the co-occurrence.

$n_X \leftarrow GoogleHit(\text{“}X\text{”})$
$n_Y \leftarrow GoogleHit(\text{“}Y\text{”})$
$n_{X \wedge Y} \leftarrow GoogleHit(\text{“}X\ Y\text{”})$
$r_{X,Y} \leftarrow CoocFunction(n_X, n_Y, n_{x \wedge y})$
return $(r_{X,Y})$

Fig. 5. Measure co-occurrence using *GoogleHit*.

Algorithm 3.2: GOOGLECOOCTOP(X, Y, k)

comment: Given person names X and Y, return the co-occurrence.

$D_X \leftarrow GoogleTop(\text{“}X\text{”}, k)$
$D_Y \leftarrow GoogleTop(\text{“}Y\text{”}, k)$
$n_X \leftarrow NumEntity(D_X \cup D_Y, X)$
$n_Y \leftarrow NumEntity(D_Y \cup D_Y, Y)$
$n_{X \wedge Y} \leftarrow NumCooc(D_X \cup D_Y, X, Y)$
$r_{X,Y} \leftarrow CoocFunction(n_X, n_Y, n_{X \wedge Y})$
return $(r_{X,Y})$

Fig. 6. Measure co-occurrence using *GoogleTop*.

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)

Fig. 7. Extract social network using *GoogleCooc*.

Algorithm 3.4: EXPANDPERSON(X, k)

comment: Extract person names from the retrieved pages.

$D \leftarrow GoogleTop(\text{“}X\text{”}, k)$
$E \leftarrow ExtractEntities(D)$
return (E)

Fig. 8. Expand person names.

than the simpler measure, the matching coefficient. However, it presents a problem: when the hit counts of two persons are greatly different, the value can not be large. For example, a PhD student (denoted as X) typically has much lower hit counts (say one tenth) than a professor (denoted as Y) with whom he co-authors. In this case, the Jaccard coefficient takes, at most, 0.1, where X always appear with Y. although we can think that X has a very strong relation with Y. A view of Fig. 4 is illustrative, where we use the overlap coefficient with a threshold. The problem is fixed because the overlap coefficient takes a minimum value of n_X and n_Y; in the example, the value of overlap coefficient between X and Y is as high as 1.0. Without a threshold, there are many coincidental relations recognized as found in Fig. 3. (Say the name of X appears only in one web page, where name Y coincidentally co-occurs.) After sufficient trial and error, we eventually employ

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

We set $k = 30$ for the JSAI case. An alternative, which we did not use, is to add some values both to denominator and numerator rather than setting a threshold; it can be considered as a smoothing technique on (low) probability estimation.

From a different perspective, this issue is related to *asymmetric measures* of similarity. Some examples are that we can set a co-occurrence measure as $n_{X\wedge Y}/n_Y$ from Y to X. This value has a clear interpretation in that it is the probability of a document including X given Y. An asymmetric measure creates a directed graph. However, mainly for visualization and analysis purposes, undirected graphs are sometimes more useful. The overlap coefficient can be considered as choosing a smaller value between two nodes as representative, i.e., $n_{X\wedge Y}/n_Y$ and $n_{X\wedge Y}/n_X$.

Pseudocode that measures the co-occurrence of two persons is shown in Fig. 5. In this paper, we define two functions using pseudocodes.

- *GoogleHit*: it returns the number of hits retrieved by a given query, and
- *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. Although we use Google as a search engine, other major search engines are also applicable platforms. Actually, our recent study investigates differences depending on search engines to be used for social network mining, and reports a slight preference for results from Google over those of the Yahoo! and MSN search engines [30]. Nevertheless, the difference is minor and we would emphasize here that we do use Google, but not exclusively, throughout the paper.

An alternative means exists to measure co-occurrence using a search engine, i.e., to use top retrieved documents, shown in Fig. 6. *NumEntity* returns the number of mentions in a given document set. *NumCooc* returns the numbers of co-occurrence of mentions in a given document set. We can use different levels of co-occurrences: document, paragraph, sentences or even line. Some related works use 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. An interesting recent algorithm categorized as *GoogleCoocTop* is the double-checking algorithm [31]: The strength of X to Y and strength of Y to X is double-checked; if both are high, the relation is recognized.

A social network is obtained using the algorithm presented in Fig. 7. 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 cases where 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. 8, 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 a previously described algorithm.

We show the extracted social network among JSAI researchers in Fig. 9. The network includes 266 nodes and 690 edges. The nodes are selected from among 1560 researchers who participated JSAI annual conferences in 2003 and the past three years; we exclude those whose hit count is less than 30, and those who do not have any edges recognized (i.e. isolated nodes). These network data are used in POLYPHONET for navigation of research presentation and retrieval of researchers at the annual conferences, as described in Section 5, and also for analysis of the network structure in order to understand the dynamics of JSAI community [32].

Some discussions are worth mentioning: First, some parameters are automatically determined if the training data are prepared. For example, the threshold (and co-occurrence measure itself) can be chosen so that the correct pairs are extracted effectively (e.g., by maximizing the F1 value on the training data). This direction of study is conducted recently and shows promising results [18]. Second, our evaluation of the co-occurrence measure is based on co-authorship probability. Although this seems good in the case of researchers (because co-authorship is the dominant relationship among researchers), more elaborate study is possible using other actual relationships such as same-affiliation or co-participation to the project: As shown below, we can extract other kinds of relationships aside from co-authorship from the Web.

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 [12,33–35]. In addition, the natural language community has specifically addressed name disambiguation as a class of word sense disambiguation [36,37].

Bekkerman and McCallum uses probabilistic models for the Web appearance disambiguation problem [12]: 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 [38]. 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*

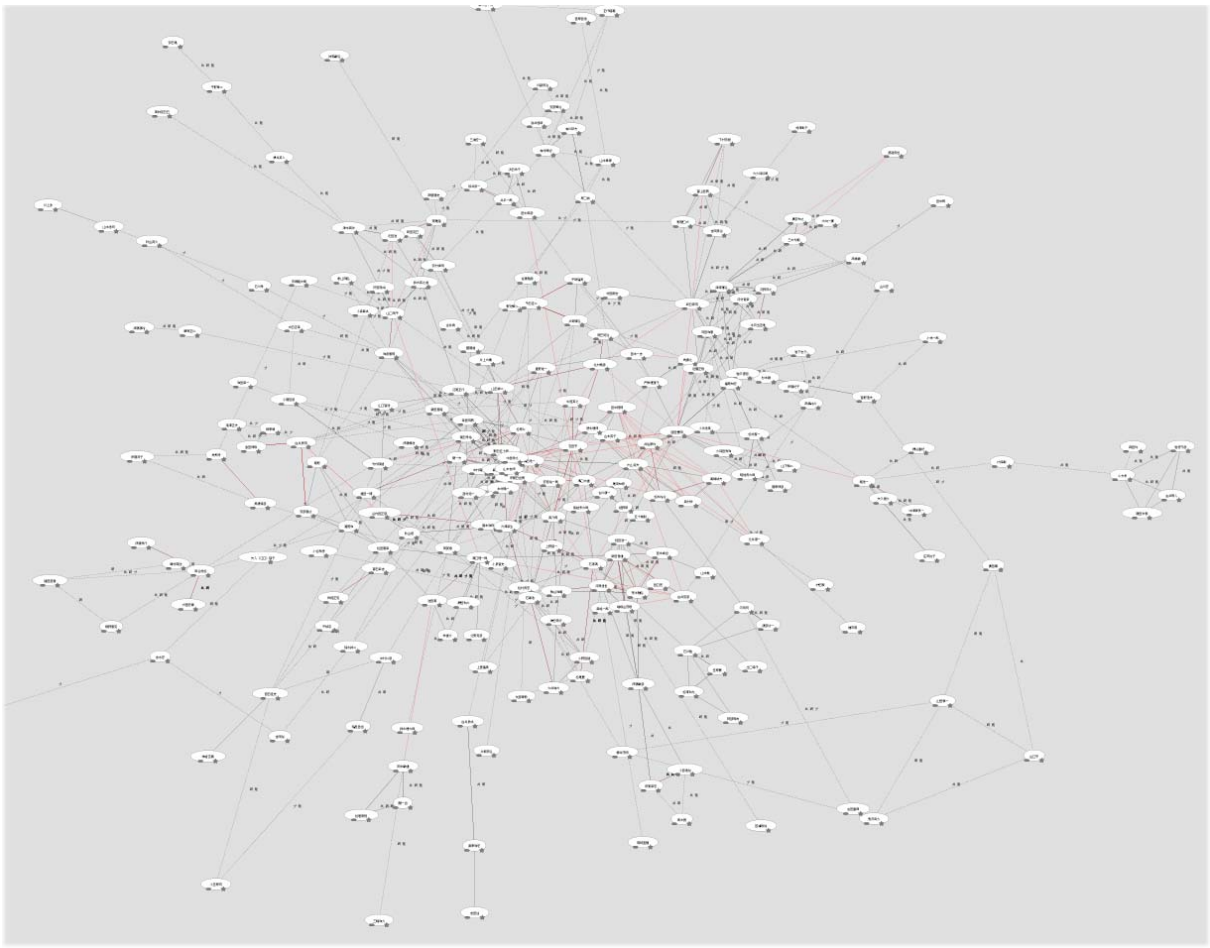

Fig. 9. A social network of JSAI researchers.

Algorithm 3.5: GOOGLECOOCCONTEXT(X, Y, W_X, W_Y)

comment: Given X, Y and word(s) W_X, W_Y, return co-occurrence.

$n_X \leftarrow GoogleHit(\text{“}X\ W_X\text{”})$

$n_Y \leftarrow GoogleHit(\text{“}Y\ W_Y\text{”})$

$n_{X \wedge Y} \leftarrow GoogleHit(\text{“}X\ Y\ W_X\ W_Y\text{”})$

$r_{X,Y} \leftarrow CoocFunction(n_X, n_Y, n_{X \wedge Y})$

return $(r_{X,Y})$

Fig. 10. Measure co-occurrence with disambiguation.

Ontology for that purpose.

In the UbiComp case, we develop a name-disambiguation module [39]. 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. 10, which is slightly modified from *GoogleCooc*. We regard keyphrases to be added as a context of a person.

Another issue, as a flip side of the same problem, is the alias problem: a person's name might have more than one surface form. Both the alias problem and the namesake problem are related to correspondence between the entity and its representations. Thinking about this, we can post deeper questions such as "what is being the same?" and "what are identities of an entity?" (c.f. [**?**]). As a pragmatic approach, several attempts have been made to solve the alias problem so far [**?**]. We will revisit the issue from a network point of view in Section 6.

4. ADVANCED EXTRACTION

This section introduces novel and useful 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 [40] defines

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 (GroFFive(B)=yes & GroFFive(C) = no)
	or (FreqX = more_than_two & FreqY = more_than_two & GroFFive(A) = yes & GroFFive(D)=no)
Proj.	(SameLine=no & GroTitle(A)=no & GroFFive(F)=yes)
	or GroTitle(C) = yes
Conf.	(GroTitle(A)=no & GroFFive(B)=no & GroFFive(D)= yes)
	or (GroFFive(A)=no & GroFFive(D)=no & GroFFive(E)= yes)

Algorithm 4.1: CLASSIFYRELATION(X, Y, k)

comment: Given person names X and Y, return the class of relation.

$D_{X \wedge Y} \leftarrow GoogleTop(\text{“}X\ Y'', k)$
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 $(class)$

Fig. 11. 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.

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)

Table 5
Evaluation by questionnaire for extracted relations (with more than 0.11 of the overlap coefficient).

class	precision	recall
Co-author	89.0% (81/91)	32.1% (81/252)
Lab	78.3% (72/92)	18.7% (72/385)
Project	50.0% (9/18)	3.0% (9/300)
Conf.	79.5% (35/44)	6.5% (35/538)

Table 6
Evaluation by questionnaire for every extracted relation.

class	precision	recall
Co-author	78.5% (135/172)	53.6% (135/252)
Lab	55.6% (109/198)	28.3% (109/385)
Project	20.3% (60/296)	20.0% (60/300)
Conf.	39.9% (222/556)	41.3% (222/538)

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. Although the previous section explained how to measure the strength of relation, 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 commu-

nities exist often and have been investigated in social network analyses [41]. It also provides interesting research topics recently in the context of complex networks to find overlapping communities [42].

Through POLYPHONET, we target the relations in a researcher community. Among them, four kinds of relation are picked up because of the feasibility of 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 Table 1 shows that we extract features from the content of each page. Attributes NumCo, 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 11 shows the pseudocode to classify relations. The *Classifier* indicates any one classifier used in machine learning such as Naive Bayes, maximum entropy model or support vector machine. In the JSAI case, we use a decision tree generating algorithm, C4.5 [43] as a classifier. Using more than 400 pages to which we manually assigned the correct labels, classification rules are obtained. 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 for classifying a web page into the classes. 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 yields low recall, presumably because laboratory pages have greater variety.

We also conducted questionnaire-based evaluation because we can not always recognize the actual relations among people from the web information. Some types of relations might be easy to extract from the web, but others are not. We made an online questionnaire and sent copies of it to 141 researchers who participated JSAI2003 Conference. Of the recipients, 82 people (58%) sent us back the responses. In the questionnaire, we randomly selected 20 persons in the network for each respondent. We asked the respondent their relation to each of the 20 persons; whether or not they had written a paper with a person, whether or not the respondent belongs (or has belonged) to the same laboratory / the same group at an institute (up to the size of 30 members), whether or not the respondent participates (or has participated) the same research project, and whether or not the respondent has joined the same conference/workshop, in addition to other questions. These questions are regarded as the correct labels corresponding to Coauthor, Lab, Project, and Conf. classes. Consequently, we have $141 \times 20 = 2820$ labels for each class.

Tables 5 and 6 show the results. Table 5 presents results of an evaluation of the automatically-recognized relations with the overlap coefficient more than the threshold (as shown in Fig. 9) to the correct labels. Table 6 shows results of an evaluation of all the recognized relations, irrespective of the overlap coefficient. It is readily apparent that the precision is high (80–90% for Co-author, Lab, and Conf. classes) in Table 5, but the recall is very low (sometimes less than 10%). In Table 6, more numerous relations are shown as recognized. Consequently, the recall increases but the precision decreases. About 50% of the coauthorships, 40% of co-attendence relations, and 30% of co-affiliation relations are recognized. Better examination of the results revealed that some of the errors occur because of the respondents' faults. For example, they sometimes do not remember a co-authorship relation; they regard project relation as broader than our stipulated definition. However, the performance was not as good as Table 4.

The difference of the results that is apparent between Table 4 and 5 is suggestive. We can recognize it as a decent probability if a source of information about the relation exists on the web. However, the information on the web does not address all existing relations, which constrains the performance of our algorithm. Therefore, we must be careful about application of the extracted network; it is useful for suggesting the existing relations, but it can not be used to verify that the two persons have no relation; it is useful to probable persons with high centrality, but we can not conclude that a person is peripheral (assuming that the number of missing links is sufficiently small).

We can improve or expand the algorithm in various ways. Below, we describe some of them. Obtaining the class of a relationship is reduced to a text categorization problem. We can employ any algorithm for the problem. Many research efforts have specifically addressed text categorization; advanced algorithms might improve the performance. For example, using unlabeled data also improves categorization [44]. One difficulty of the text categorization is the cost to annotate the correct label for training data. Therefore, the algorithm surely is helpful.

Relationships depend on the target domain. Consequently, we must define classes to be categorized depending on a domain. A possible expansion is to use an unsupervised categorization (or clustering) algorithm to recognize the types of relations. We first fetch pages related to pairs of persons using *GoogleTop*, extract attributes for each page, and apply a clustering algorithm. Then, we look into the resultant clusters, and annotate labels for clusters if they are understandable and meaningful. This extension is proposed and evaluated in our recent work [45] by targeting the politician-region domain as well as the researcher domain.

Another expansion is to make the algorithm more scalable. Vastly numerous pages exist on the Web. For that reason, the *ClassifyRelation* module becomes inefficient when k is large. Especially to seek minor relations between two entities, top-ranked Web pages are not always useful. 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 rela-

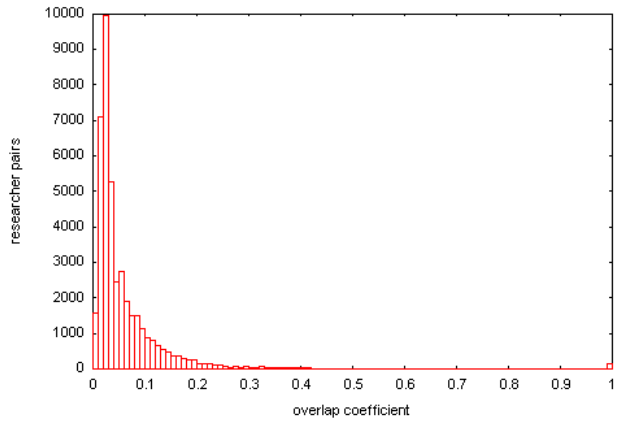

Fig. 12. Number of pairs versus the overlap coefficient.

tions, 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 [18].

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 arise from 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. 12. 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. 13. 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, ${}_{503}C_2 = 126253$ 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. 14. 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 performance improves as k increases. (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 as candidates after filtering. Although m

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
  - $D \leftarrow GoogleTop(\text{“}X'', k)$
  - $E \leftarrow ExtractEntities(D)$
  - **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)

Fig. 13. Extract social networks in a scalable way.

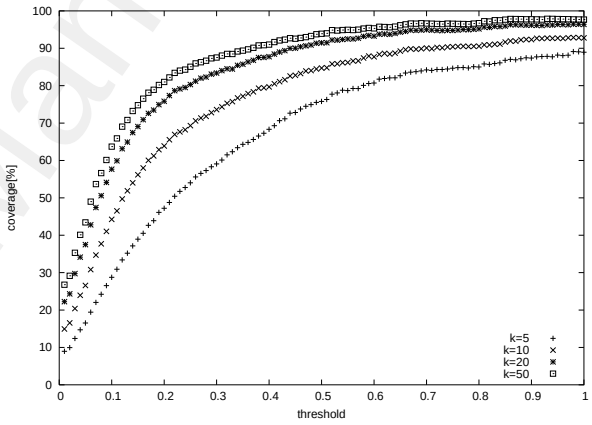

Fig. 14. Coverage of *GetSocialNetScalable* for JSAI case.

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 $n = 7000$.[8]

We show an example of the large-scale network extracted. Figure 15 is a network for 2,879 researchers in information engineering and science in Japan, by the hierarchical category on a researcher database[9]. About 1000 nodes are included in the displayed network. To extract the network, 4,142,881 queries were necessary if we used the *GetSocialNet* module, but actually, 137,967 queries were made using a *GetSocialNetScalable* module, which saved 97% of the queries.

Although the *GetSocialNetScalable* module works well in the scope of our experiment, there might be some limitation of the scalability: Assuming that X has a huge number of mentions on the Web, it is difficult to filter out relations to other

[8] In case of the disaster mitigation research community in Japan.
[9] http://read.jst.go.jp/index_e.html

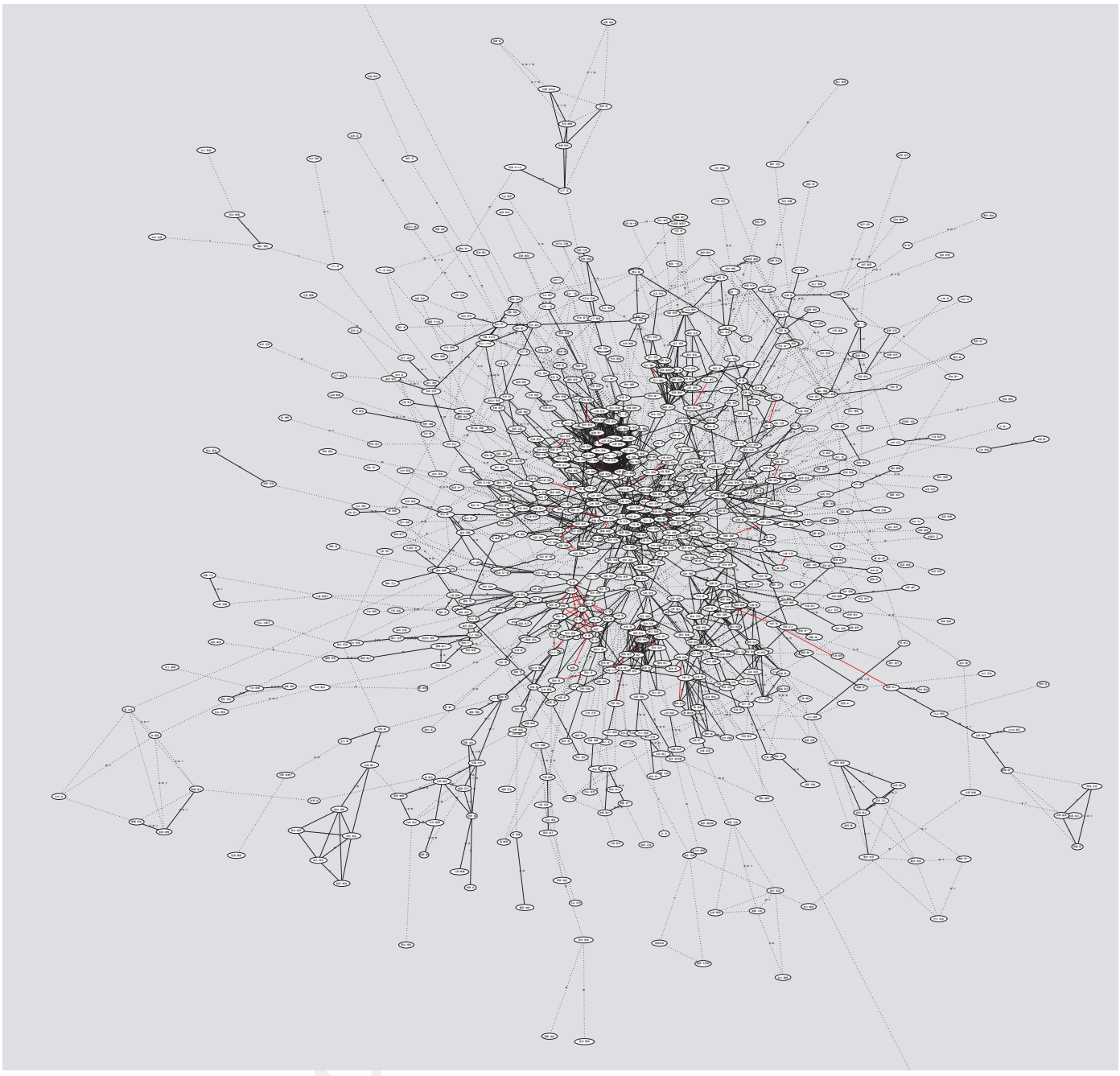

Fig. 15. Network of information engineering and information science researchers in Japan.

entities by merely looking at k top documents. Theoretically, the performance would be degraded if the Web were larger with constant k. We might need to produce more elaborate query construction for efficient filtering, which necessitates further investigation.

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 [11].

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

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)

Fig. 16. Extract keywords for a person.

Algorithm 4.4: CONTEXTSIM(X, Y, W_L)

comment: Given names X and Y, and word list W_L, return the similarity.

for each $W \in W_L$
  do $a_W \leftarrow GoogleCooc(X, W)$
  do $b_W \leftarrow GoogleCooc(Y, W)$
$s_{X,Y} \leftarrow$ similarity of two vectors $a = \{a_W\}$ and $b = \{b_W\}$
return $(s_{X,Y})$

Fig. 17. Measure the context similarity of two persons.

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

Fig. 18. Exemplary keywords for *Dan Brickley*.

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. 16. Collecting documents retrieved by a person name, we obtain a set of words and phrases as candidates for keywords. We use Termex [46] 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 18 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 7 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 $k_1 = 10$ 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 [47].) 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.

Table 7
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 ... W_m
X_1	...
X_2	...
X_3	... ...
...	...
X_n	...

	X_1 X_2 ... X_M
X_1	...
X_2	...
X_3	...
...	... ...
X_n	...

Fig. 19. Affiliation matrix and adjacent matrix.

4.3.2. *Affiliation network*

Co-occurrence information of words and persons forms a matrix. Figure 19 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* whereas a person-person matrix is called an *adjacent matrix* [41]. Figure 20 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 17 presents 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

	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	...
...	...	...	...	...	...	...

Fig. 20. Example of a person-to-word co-occurrence matrix.

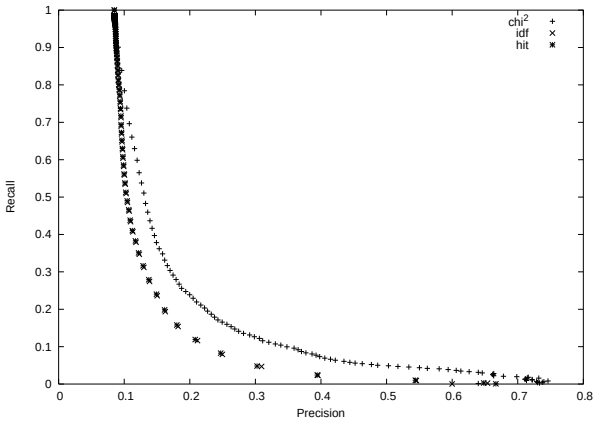

Fig. 21. Precision and recall for session identification.

with many person names. Therefore, statistical methods are effective: we first apply χ^2 statistics to the affiliation matrix and calculate cosine similarity [48].

One evaluation is shown in Fig. 21. 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* [41]. 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.

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 [10] and for JSAI2005 [11].

Table 8
Number of participants at conferences.

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

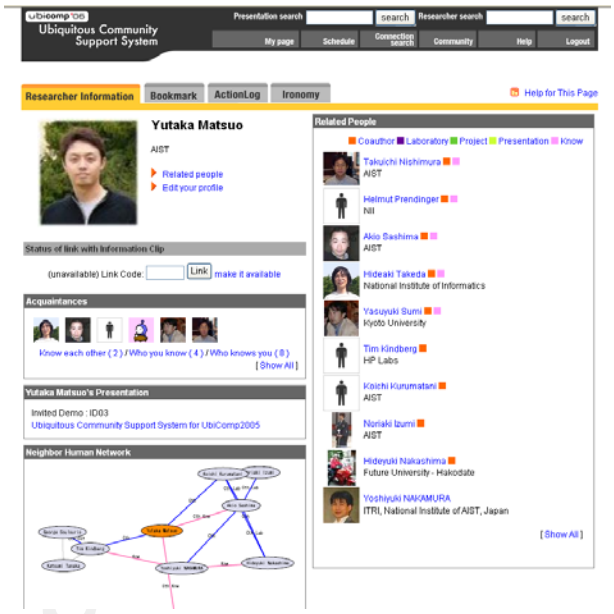

Fig. 22. My page on POLYPHONET.

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 [49] and a location information display system [50] in the ubiquitous computing environment at the conference sites.

Figure 22 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. The system can output a FOAF file for each participant. Figure 23 shows the shortest obtained path between two persons on a social network. Figure 24 is a screenshot that illustrates when three persons come to an 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 8. Comments were almost entirely positive; they enjoyed using the system.

POLYPHONET provides an area for future work in which feedback is collected through the system. That information is

[10] http://www.ubicomp-support.org/ubicomp2005/.
[11] http://jsai-support-wg.org/polysuke2005/.

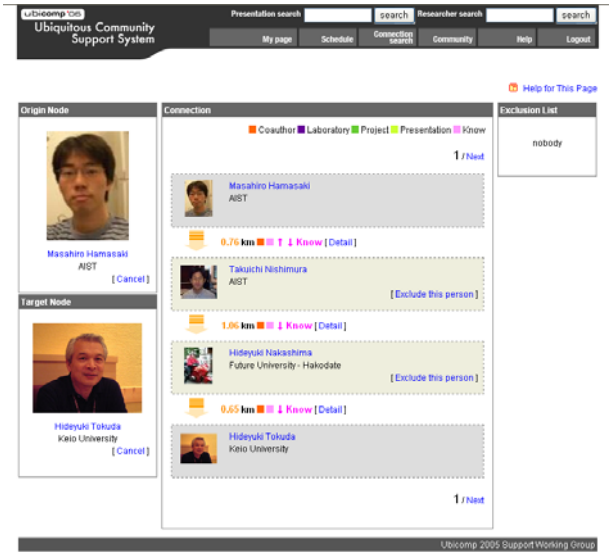

Fig. 23. Shortest path from a person to a person on POLYPHONET.

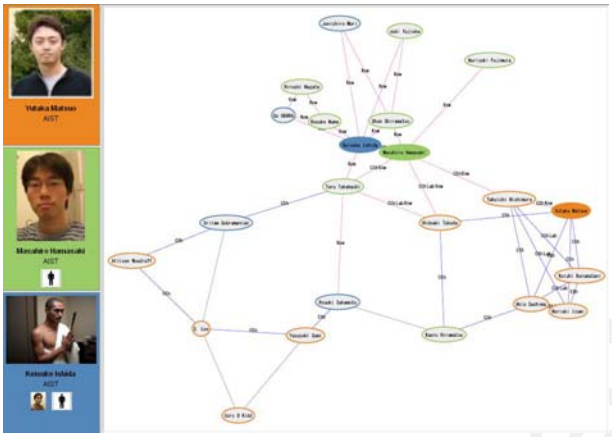

Fig. 24. Social network among three persons on POLYPHONET.

useful as training data if a user registers another user as an acquaintance. In this way, the system would improve over time in the style of active learning, which can be sought in the future.

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.

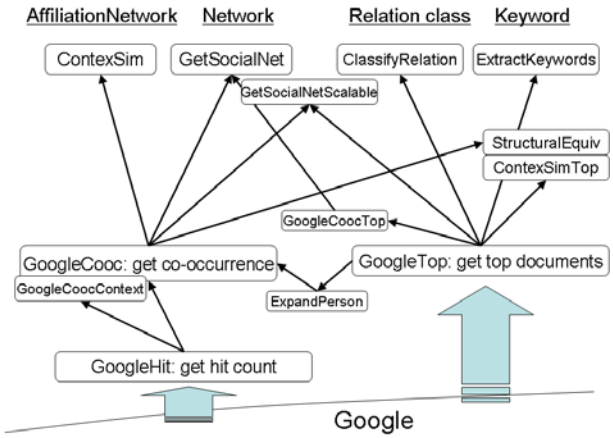

Fig. 25. Overview of module dependency.

According to that concept, we propose a new architecture to extract a social network from the Web, called *Iterative 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 [12] 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.

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 [51]. 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 Iterative 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 [52] will benefit our research greatly if it actually scales properly.

Figure 25 shows an overview of the module dependencies we described in this paper. *GoogleHit* and *GoogleTop* are remarkably versatile yet simple modules. We note that two modules are

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

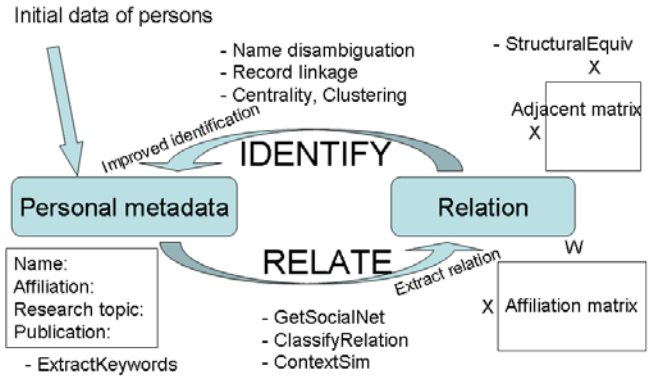

Fig. 26. Relate-Identify process of *Iterative Social Network Mining*.

not introduced 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 [53]. The *StructuralEquiv* module calculates structural equivalence, which plays an important role in the Relate-Identify process.

Figure 26 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 described 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 [54]. 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 for the Relate-Identify process is not implemented in POLYPHONET, we have partially realized the process, 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 for 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 Iterative 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 to many knowledge-based systems in the future. Acquiring knowledge through Googling is a similar concept to ours [55]. 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 the New Energy and Industrial Technology Development Organization (NEDO), No. 04A11502a.

References

[1] S. Staab, P. Domingos, P. Mika, J. Golbeck, L. Ding, T. Finin, A. Joshi, A. Nowak, R. Vallacher, Social networks applied, IEEE Intelligent Systems (2005) 80–93.

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

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

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

[5] P. Mika, Web semantics: Science services and agents on the world wide web (2007) 5–15.

[6] L. Adamic, O. Buyukkokten, E. Adar, Social network caught in the web, First Monday 8 (6).

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

[8] T. Miki, S. Nomura, T. Ishida, Semantic web link analysis to discover social relationship in academic communities, in: Proc. SAINT 2005, 2005.

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

[10] C. Ramakrishnan, K. Kochut, A. Sheth, A framework for schema-driven relationship discovery from unstructured text, in: Proc. ISWC2006, 2006.

[11] A. Culotta, R. Bekkerman, A. McCallum, Extracting social networks and contact information from email and the web, in: CEAS-1, 2004.
[12] R. Bekkerman, A. McCallum, Disambiguating web appearances of people in a social network, in: Proc. WWW 2005, 2005.
[13] M. Harada, S. Sato, K. Kazama, Finding authoritative people from the web, in: Proc. Joint Conference on Digital Libraries (JCDL2004), 2004.
[14] C. Faloutsos, K. S. McCurley, A. Tomkins, Fast discovery of connection subgraphs, in: Proc. ACM SIGKDD 2004, 2004.
[15] P. Knees, E. Pampalk, G. Widmer, Artist classification with web-based data, in: Proc. 5th International Conf. on Music Information Retrieval (ISMIR), 2004.
[16] P. Turney, Mining the web for synonyms: PMI-IR versus LSA on TOEFL, in: Proc. ECML-2001, 2001, pp. 491–502.
[17] E. Terra, C. Clarke, Frequency estimates for statistical word similarity measures, in: Proc. HLT/NAACL 2003, 2003.
[18] Y. Jin, Y. Matsuo, M. Ishizuka, Extracting social networks among various entities on the web, in: Proc. ESWC 2007, 2007.
[19] Y. Matsuo, J. Mori, M. Hamasaki, H. Takeda, T. Nishimura, K. Hasida, M. Ishizuka, POLYPHONET: An advanced social network extraction system, in: Proc. WWW 2006, 2006.
[20] P. Cimiano, S. Handschuh, S. Staab, Towards the self-annotating web, in: Proc. WWW2004, 2004, pp. 462–471.
[21] P. Cimiano, G. Ladwig, S. Staab, Gimme´ the context: Context-driven automatic semantic annotation with cpankow, in: Proc. WWW 2005, 2005.
[22] T. Calishain, R. Dornfest, Google Hacks: 100 Industrial-Strength Tips & Tools, O'Reilly, 2003.
[23] T. Finin, L. Ding, L. Zou, Social networking on the semantic web, The Learning Organization.
[24] B. Aleman-Meza, M. Nagarajan, C. Ramakrishnan, A. Sheth, I. Arpinar, L. Ding, P. Kolari, A. Joshi, T. Finin, Semantic analytics on social networks: Experiences in addressing the problem of conflict of interest detection, in: Proc. WWW2006, 2006.
[25] J. Goecks, E. D. Mynatt, Leveraging social networks for information sharing, in: Proc. ACM CSCW 2004, 2004, pp. 328–331.
[26] J. Mori, M. Ishizuka, T. Sugiyama, Y. Matsuo, Real-world oriented information sharing using social networks, in: Proc. ACM GROUP'05, 2005.
[27] R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, The web and social networks, IEEE Computer 35 (11) (2002) 32–36.
[28] C. Manning, H. Schütze, Foundations of statistical natural language processing, The MIT Press, London, 2002.
[29] Y. Matsuo, H. Tomobe, K. Hasida, M. Ishizuka, Finding social network for trust calculation, in: Proc. 16th European Conference on Artificial Intelligence (ECAI2004), 2004, pp. 510–514.
[30] Y. Matsuo, T. Hironori, T. Nishimura, Robust estimation of google counts, in: Proc. AAAI 2007, 2007.
[31] H. Chen, M. Lin, Y. Wei, Novel association measures using web search with double checking, in: Proc. ACL2006, 2006.
[32] Y. Yasuda, Y. Matsuo, Making invisible colleges visible – ai researchers' network positions and productivity, in: Proc. International Sunbelt Social Network Conference (Sunbelt XXVI), 2006.
[33] R. Guha, A. Garg, Disambiguating entities in web search, tAP project, http://tap.stanford.edu/PeopleSearch.pdf.
[34] L. Lloyd, V. Bhagwan, D. Gruhl, A. Tomkins, Disambiguation of references to individuals, Tech. Rep. RJ10364(A0410-011), IBM Research (2005).
[35] B. Malin, Unsupervised name disambiguation via social network similarity, in: Workshop Notes on Link Analysis, Counterterrorism, and Security, 2005.
[36] N. Wacholder, Y. Ravin, M. Choi, Disambiguation of proper names in text, in: Proc. 5th Applied Natural Language Processing Conference, 1997, pp. 202–208.
[37] G. S. Mann, D. Yarowsky, Unsupervised personal name disambiguation, in: Proc. CoNLL, 2003.
[38] X. Li, P. Morie, D. Roth, Semantic integration in text: From ambiguous names to identifiable entitiesb, AI Magazine Spring (2005) 45–68.
[39] D. Bollegala, Y. Matsuo, M. Ishizuka, Disambiguating personal names on the web using automatically extracted key phrases, in: Proc. ECAI 2006, 2006.
[40] I. Davis, E. V. Jr., RELATIONSHIP: A vocabulary for describing relationships between people, http://vocab.org/relationship/.
[41] S. Wasserman, K. Faust, Social network analysis. Methods and Applications, Cambridge University Press, Cambridge, 1994.
[42] G. Palla, I. Derenyi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (2005) 814.
[43] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, California, 1993.
[44] K. Nigram, A. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using em, Machine Learning 39 (2000) 103–134.
[45] J. Mori, T. Tsujishita, Y. Matsuo, M. Ishizuka, Extracting relations in social networks from web using similarity between collective context, in: Proc. ISWC 2006, 2006.
[46] H. Nakagawa, A. Maeda, H. Kojima, Automatic term recognition system termextract, http://gensen.dl.itc.utokyo.ac.jp/gensenweb_eng.html.
[47] J. Mori, Y. Matsuo, 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.
[48] S. Chakrabarti, Mining the Web: Discovering Knowledge from Hypertext Data, Morgan Kaufmann, 2002.
[49] M. Hamasaki, H. Takeda, I. Ohmukai, R. Ichise, Scheduling support system for academic conferences based on interpersonal networks, in: Proc. ACM Hypertext 2004, 2004.
[50] T. Nishimura, Y. Nakamura, H. Itoh, H. Nakamura, System design of event space information support utilizing CoBITs, in: Proc. ICDCS 2004, 2004, pp. 384–387.
[51] A. Anagnostopoulos, A. Z. Broder, D. Carmel, Sampling search-engine results, in: Proc. WWW 2005, 2005, pp. 245–256.
[52] M. Cafarella, O. Etzioni, A search engine for natural language applications, in: Proc. WWW2005, 2005.
[53] M. Sahami, T. Heilman, A web-based kernel function for matching short text snippets, in: International Workshop on Learning in Web Search (LWS2005), 2005, pp. 2–9.
[54] R. S. Burt, Structural Holes: The Social Structure of Competition, Harvard University Press, Cambridge, MA, 1992.
[55] P. Cimiano, S. Staab, Learning by googling., SIGKDD Explorations 6 (2) (2004) 24–33.