Entity Information Extraction from the Web using Search Engine: Methodology and Application
検索エンジンを利用したウェブからのエンティティ情報抽出手法と応用に関する研究

A DISSERTATION
SUBMITTED TO THE GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
AND THE COMMITTEE ON GRADUATE STUDIES
OF UNIVERSITY OF TOKYO
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Junichiro MORI

June 2007

Abstract

The current development of Web applications such as Blogs and Wiki enables users to easily create and disseminate their contents in the Web. As the contents on the Web are rapidly growing, the quantity of information is recently becoming more important in the Web. With the large quantity of information, the Web has now turned to the huge corpus that can be easily accessible using search engines, that opens new possibility to handle the vast relevant information and mine important structures and knowledge.

In addition to the trend of "Web as corpus", another important aspect of the current Web is that our daily life is reflected in the Web. For example, social networking services (SNSs) where users maintain an online network of friends have recently received much attention on the Web. Information about tens of millions of people and their relationships are currently available in the Web. Communication and information sharing in the real world are also reflected in the Web. As users publish their daily activities and communicate in the Web, the Web is now becoming another form of our society.

With the current trends of "Web as Corpus" and "Web as Society", the large amount of information that are originated from our daily activities in the real world are available on the Web. On top of these trends in the Web, there is a new requirement for information retrieval that users try to find the "entity-based" information rather than documents. Here, entity is defined as the object in the real world such as person, location, and organization. In addition to single entity information, relation information among entities from the Web are also becoming important information to be retrieved by users.

Users are currently searching for entity-based information and entity relations on top of existing document-based Web information.The Semantic Web is one approach to realize the entity-based information retrieval. In the Semantic Web, every resource is annotated with metadata using ontology. Users can easily search and find the entity-based information using the annotated metadata. However, because data should be annotated with metadata in advance to fully use the Semantic Web technologies, there is a major problem of metadata annotation in the Semantic Web. Therefore, there is still a huge gap between the Semantic Web and the current Web where most data are unstructured.

Aiming at realizing information services based on entity-based information and entity relations as a next stage of current information retrieval, in this thesis we propose methods for extracting entity information and entity relations from the Web. The key features of our approach are to leverage existent search engine and obtain several Web-scale statics such as hit counts and snippets in order to assess entity-related information. We construct the entity model using the information obtained from search engine. Applying several text processing techniques such as term weighting, similarity measure, and clustering to the entity model, our methods extract entity information, entity relations and social networks. The extracted information can be applied to several applications. We first develop the researcher search system in which the information about researchers and relationships are automatically extracted from the Web. We also develop the information sharing system and the expert finding system using the extracted social networks.

Overall, in this thesis we address two research questions for extracting entity information from the Web: (1) how search engines can be used to access the Web corpus and extract entity information from the Web and (2) how the extracted entity information can be used to support users in entity-based information services.

For first question, we first propose a method for constructing the entity model using the information obtained from search engine. We then propose a method of keyword extraction for extracting entity information from the Web. The proposed method is based on the statistical features of word co-occurrence obtained from search engine. We also propose a method that extracts descriptive labels of relations among

entities automatically such as affiliations, roles, locations, part-whole, social relationships. Fundamentally, the method clusters similar entity pairs according to their collective contexts obtained from search engine. The descriptive labels for relations are obtained from the results of clustering. Finally, We propose a method that automatically extracts social networks from the Web. The method leverages co-occurrence information obtained from search engine to estimate the relation between entities.

For second question, we develop the researcher search system. The system is a Web-based system for an academic community to facilitate communication and mutual understanding based on entity information and social networks extracted from the Web. We also develop a real-world-oriented information sharing system. The system enables users to determine who has access to particular information based on the social networks and network analysis. We finally develop the expert finding system that locates relevant and socially close experts for information seekers. The system leverages the entity information and social networks of a Web community in order to find experts who have appropriate expertise.

Acknowledgements

I would like to thank, first and foremost, my advisor, Prof. Dr. Mitsuru Ishizuka, for his guidance, support, and patience through out my graduate career and during the completion of this thesis. Prof. Ishizuka always encouraged me and provided insights, suggestions, and ideas that improved all of my graduate work. Prof. Ishizuka is truly a great mentor and role model, and I am extremely fortunate to have had him as an advisor. I am also extremely grateful to the members of my committee, Prof. Dr. Jun Adachi, Prof. Dr. Masaru Kitsuregawa, Prof. Dr. Shuichi Sakai, Prof. Dr. Takashi Chikayama, and Prof. Dr. Tohru Asami for their advice and comments in this thesis.

My special thanks go to Dr. Yutaka Matsuo for his advice and encouragement on this thesis. Thanks also go to Dr. Koichi Hashida for providing me the opportunity to work on the project in National Advanced Institute of Science and Technology (AIST), to Prof. Dr. Boi Faltings for providing me the opportunity to study in Ecole Polytechnique Fédéral de Lausanne (EPFL) for one year, to Dr. Alexander Kroñer, Prof. Dr. Anthony Jameson, and Prof. Dr. Woflgang Wahlster for providing me the opportunity to work on the SharedLife project in DFKI. I am also grateful to the members of Event Space Information Support Project (ESISP), Dr. Takuichi Nishimura, Dr. Yoshiyuki Nakamura, Prof. Dr. Hideaki Takeda, Kesuke Ishida, Dr. Masahiro Hamasaki, Dr. Ikki Omukai, Dr. Kosuke Numa for giving me the opportunity to apply my research results to the JSAI Web system.

I would like to acknowledge the students and other staff members in Ishizuka Laboratory, especially the students in "Matsuo Gumi", Dr. Naoaki Okazaki, Danushka Bollegala, Daisuke Kobayashi, Sakaki Takeshi, Yingzi Jin, Nguyen Phuoc Tac Dat,

Takumi Tsujisita, Kenji Hirohata, Jun Karamon, Shinichiro Minotsu, and Jo Okajima for having discussion and giving advice.

Last but not the least, I would like to thank my parents, my grand parents, and my brothers for their great love and supporting me in my educational pursuits.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Current Web: from Quality to Quantity

The current development of Internet infrastructure such as broadband and wireless network enables users to easily access the Web. Nearly 87 million people [1] in Japan are currently using Internet. Moreover, the current development of Web applications enables users to easily create and disseminate their contents in the Web. For example, using Blogs which are diary-like sites including multimedia contents such as photos and videos, users can easily publish their information. Nearly 8.68 million people [2] in Japan are currently using Blog services.

With the rapidly growing contents on the Web, the recent Web has witnessed the transition from quality to quantity of information. A few years ago, when people tried to find information in the Web, they relied on the several "authority" sites that aggregate and disseminate valuable information. The algorithms for ranking the Web sites such as HITS and Pagerank have been developed and applied to find such sites. However, the recent information explosion and distribution where users can easily publish their information on the Web has made it difficult to find valuable information only by using such hub and authority-based algorithms. As the contents on the Web are rapidly increasing, the quantity of information is recently becoming more important in the Web.

Web as Corpus

The importance of quantity of information has been explained with recent "collective intelligence" in the Web. Collective intelligence is the capacity of communities to co-operate intellectually in creation, innovation and invention. For example, Wikipedia, an online encyclopedia is based on the notion that every user can add an entry, is a successful site using the idea of the collective intelligence. Folksonomy, a style of

[1] The number is based on the survey of Japan Ministry of Internal Affairs and Communications in 2006.

[2] The number is based on the survey of Japan Ministry of Internal Affairs and Communications in 2006.

collaborative categorization of Web sites using freely chosen keywords (or tags), is another example of the collective intelligence. As seen in Wikipedia, every single user contributes to creating large quantity of information and then as seen in Folksonomy, the information is organized and guided by user communities.

The collective intelligence is also emerging in huge language resources of the Web documents that contain hundreds of billions of words of text. Therein, search engine plays an important role to access the resources. The simple way to access the language resources in the Web is to leverage hit counts of search engine as word frequencies. For example, when checking the spell, speculater or speculator, Google gives 4,700 for the former and 1,210,000 for the latter. As seen in this example, the collective intelligence of majority decision in the Web can be easily obtained simply by exploiting Google hit counts. With the large quantity of information, the Web has turned to the huge corpus that can be easily accessible source of language material using search engines, which in turn opens new possibility to handle the vast relevant information and mine important structures and knowledge.

Web as Society

In addition to the trend of "Web as corpus", another important aspect of the current Web is that our daily life is reflected in the Web. For example, social networking services (SNSs) have recently received considerable attention on the Web. SNSs enable users to maintain an online network of friends or associates for social or business purposes. Therein, the users can create their contents such as profiles and Blogs and communicate with their friends. Information about tens of millions of people and their relationships are published in several SNSs. For example, more than 10 million users [3] are using mixi, the largest SNS in Japan.

As users publish their daily activities and social relationships in Blogs and SNSs, the Web is currently reflecting the information in the real world and the information is constantly updated through the contents that the users create online. Communication and information sharing in the real world are also reflected in the Web. Using

[3]The number based on the survey of mixi, inc in May 2007.

several communication tools such as Email, Instant Messenger, and SNSs, users can communicate each other and share information online as they do in the real world. As information and communication in the real world have been reflected in the Web. The Web is becoming another form of our society.

Information Retrieval: from Document to Entity

With the current trends of "Web as Corpus" and "Web as Society", the large amounts of information that are originated from our daily activities in the real world are available on the Web. On top of these trends in the Web, there is a new requirement for information retrieval that users try to find the "entity-based" information rather than documents. Here, entity is defined as the object in the real world such as person, location, and organization. Large amounts of information about people, places and other entities are currently available in the Web. To extract and deduce information about these named entities has many practical applications. For example, if you are browsing the news site it would be interesting to be able to click on a name and get information about the entity associated with that name. A user might also be interested in finding people related to the entity. In addition to single entity information, as we can see the recent trend of social networks which are basically representing the structure of relations among entities, relation information among entities from the Web (e.g. relation between two persons or relation between a person and an organization) are also becoming important information to be retrieved by users.

The entity-based information retrieval should exist on top of current document retrieval. For example, when a user wants to know "Prof. Mitsuru Ishizuka", he might put the query "Mitsuru Ishizuka" into a search engine and try to find the information about Prof. Ishizuka from the search results. Therein, the final goal of the user is not to find the documents that include descriptions about Prof. Ishizuka but to find the related information of Prof. Ishizuka such as his students, research fields, affiliations, and projects. In other words, what the user wants to know is the information or attributes about Prof. Ishizuka as a person (or more precisely researcher) entity. In

order to know about him further, the user might try to find the relation between Prof. Ishizuka and his student, co-author, or colleague. The user might be also interested in the relation between Prof. Ishizuka and his affiliation. As seen in this example, users are currently searching for entity-based information and entity relations on top of existing document-based Web information.

In current information retrieval, a user has a specific information need, and the system provides a list of documents that satisfy all or parts of that information need. Typically the list is presented in an order of decreasing relevance, where relevance is determined by the system. It is often the user's job to connect the pieces of information together in order to satisfy a precise information need such as finding specific entity information. The Semantic Web is one approach to realize the entity-based information retrieval. In the Semantic Web, every resource is annotated with metadata using ontology. For example, "Prof. Ishizuka" is explicitly represented as an instance of Person class and related information about him such as affiliations and research fields are described with metadata. Users can easily search for and find the information about Prof. Ishizuka using the annotated metadata. However, because data should be annotated with metadata in advance to fully use the Semantic Web technologies, the annotation of metadata is a major problem to realize the Semantic Web. Therefore, there is still a huge gap between the Semantic Web and the current Web where most data are unstructured. Now the question is how to fill the gap between the current and the Semantic Web to realize entity-based information.

1.2 Contributions of the Thesis

Aiming at realizing information services based on entity-based information and entity relations toward a next stage of current information retrieval, in this thesis we propose the methods for extracting entity information and entity relations from the Web. The key features of our approach are to leverage existent search engine and obtain several Web-scale static such as hit counts and snippets in order to assess entity-related information. We construct the entity model using the information obtained from search engine. Applying several text processing techniques such as term weighting,

similarity measure, and clustering to the entity model, our methods extract entity information, entity relations and social networks. The extracted information can be applied to several applications that are based on the entity information. We first develop the researcher search system that the information about researchers and relationships are automatically extracted from the Web. We also develop the information sharing system and the expert finding system using the extracted social networks.

Overall, in this thesis we address two major research questions for extracting entity information from the Web: (1) how the search engine can be used to access the Web corpus and extract entity information from the Web and (2) how the extracted entity information can be used to support users in entity-based information services. Our main contributions in this thesis include:

Methods for Modeling Entity Information from the Web

We propose a basic method to construct the entity model using the information obtained from search engine. The information includes hit counts, co-occurrence, and snippets. Applying several text processing techniques such as term weighting, similarity measure, and clustering to the entity model, we develop following methods for extracting entity information, entity relations, and social networks from the Web.

Method for Extracting Entity Information from the Web using Search Engine

We propose a method of extracting entity information in form of keyword from the Web. The proposed method is based on the statistical features of word co-occurrence that are obtained from search engine. The basic idea is a following: if a word co-occurs with an entity in many Web pages, the word might be a relevant keyword about the entity. Importantly, our method extracts relevant keywords depending on the context of the entity. Our evaluation shows better performance to $tfidf$-based keyword extraction. Publications based on this research are: [53] [56] [57] [52].

Method for Extracting Entity Relations from the Web using Search Engine

We propose a method that automatically extracts descriptive labels of relations among entities automatically such as affiliations, roles, locations, part-whole, social relationships. Fundamentally, the method clusters similar entity pairs according to their collective contexts in Web documents. The descriptive labels for relations are obtained from results of clustering. The proposed method is entirely unsupervised and is easily incorporated with existing social network extraction methods. Our experiments conducted on entities in researcher social networks and political social networks achieved clustering with high precision and recall. The results showed that our method is able to extract appropriate relation labels to represent relations among entities in the social networks. Publications based on this research are: [60] [55]

Method for Extracting Social Networks from the Web

We propose a method that automatically extracts social networks from the Web. The Web is currently a huge source of information for the relation between entities. Our method leverages co-occurrence information obtained from a search engine to extract a social network among entities. The basic idea is a following: if two entities co-occurs in many Web pages, they might have a relation. We evaluated several co-occurrence measures to find the robast measure for extracting a social network from the Web. Combining several information about entity and relation that are also extracted automatically from the Web, we develop a method for extracing a social network in the way that the social network is easy to understand and applicable for practical applications. Publications based on this research are: [54] [138] [168]

Application of Social Networks: Researcher Search System, Information Sharing System, and Expert Finding System

- Researcher Search System

  We develop a researcher search system that is a Web-based system for an academic community to facilitate communication and mutual understanding based on a social network extracted from the Web. The system provides various types of retrieval are possible on the social network: researchers can be sought by name, affiliation, keyword, and tag; related researchers to a retrieved researcher are listed; and a search for the shortest path between two researchers can be made.

- Information Sharing System

  We develop a real-world oriented information sharing system that uses social networks extracted from the Web. The system automatically obtains users' social relationships by mining various external sources. It also enables users to analyze their social networks to provide awareness of the information dissemination process. Users can determine who has access to particular information based on the social relationships and network analysis.

- Expert Finding System

  We propose a method that employs the user profile and social structure of a Web community in order to find experts who have appropriate expertise and are likely to be able to reply to a information request. We addressed the issue in the scenario from the actual social network service for sharing recipes. Utilizing the user information and social structure from the existing Web community, we implemented and operated the community mining system which locates relevant and socially close experts for information seekers.

Publications based on these researches are: [59] [58]

1.3 Organisation of the Thesis

This thesis is organized as follows: In Chapter 2, we review background and related work. Then, in Chapter 3, we present a basic method for modeling entity information from the Web. In Chapter 4, we present a method for extracting entity information from the Web. Later, in Chapter 5, we present a method for extracting entity relations. We then shift our focus to how to extract social networks from the Web and apply to the information system. First, in Chapter 6, we present a basic method for extracting social networks from the Web and we also present a researcher search system using social networks. In Chapter 7, we present a information sharing system using social networks. In Chapter 8, we present an expert finding system using social networks. Finally, in Chapter 9, we conclude this thesis and describe some open questions and future research directions.

Chapter 2

Background and Related Work

In this thesis, we mainly address the extraction of entity information and social networks using a search engine from the Web. Therefore, our work touches on tree major research areas: information extraction, Semantic Web and social network. This chapter reviews related work in these areas. First, we summarize the research on information extraction, emphasizing Web mining-based techniques that are most closely related to our work (Section 2.1). Then, we review recent work on the Semantic Web and discuss its connection to information extraction (Section 2.2). Finally, we summarize relevant work on social networks, and comment on its connection to our social network extraction and applications.

2.1 Information Extraction and Web mining

Aiming at extracting entity information, our method is regarded as an IE (Information Extraction) method. Up to now, many IE methods have relied on predefined templates and linguistic rules or machine learning techniques to identify certain entities in text documents [131]. For example, some previous IE researches have addressed the extraction of entity information. In [118], the authors propose a method to extract artist information from Web pages, such as name and date of birth, and automatically generate his or her biography. In [124], they address the extraction of entity information such as name, project, publication in a specific department using unsupervised information extraction. These methods usually define properties, domains, or ontology beforehand. Many methods for extracting entity information from documents such as newspapers and scientific papers have been studied. In contrast to those documents, Web pages are too diverse and heterogeneous to apply the previous methods since they include free text and unstructured data, lack regular sentences. We extract entity information based on Web-scale statistic such as hit counts and snippets using a search engine without any predefined restrictions. We also use existing probabilistic measures such as mutual information [121] and Log-Likelihood [125] as newly defined Web-based measures.

Extracting relations between entities is related closely to existing extraction methods of social networks. Several studies have addressed extraction of social networks

automatically from various sources of information such as the Web, e-mail, and contacts [6, 46, 137, 168]. While most approaches for social network extraction have focused on the strength of the relation, few studies have addressed automatic identification of underlying relations. Matsuo et al. employed a supervised machine learning method to classify four types of relations in a research community [138].

There have also been several important works that have examined supervised learning of relation extraction in the field of natural language processing and information extraction [151, 152, 153]. However, a supervised method requires large annotated corpora, which cost a great deal of time and effort. In addition, it is necessary to gather the domain specific knowledge a priori to define extracted relations. Our method is fully unsupervised and requires no annotated corpora. Furthermore, our method works domain independently and requires no pre-defined relations. For further improvement of our method, it might be worth considering exploitation of weakly supervised and bootstrapping methods [154, 155] that rely on a small set of pre-defined initial seeds instead of a large annotated corpus.

Several studies have proposed relation extractionfrom a large language corpus using a bag-of-words of context [68, 33]. Our method can be considered as an application of relation extraction methods in NLP to social networks and a Web mining. We are aiming at easily incorporating into extraction methods of social network from the Web. Therefore, our method uses context information that is obtained during extraction of social networks. Consequently, it serves to enrich such networks by adding relation labels.

Aiming at extraction of the relation labels in automatically extracted social network from the Web, our method is a Web mining method. Recent approaches of Web mining toward the Semantic Web use the Web as a huge language corpus and combine it with a search engine. This trend is observed not only in recent social network extraction [137, 138] but also in ontology population for entities [165, 166] and relations [160, 167]. The underlying concept of these methods is that it uses globally available Web data and structures to annotate local resources semantically to bootstrap the Semantic Web. In line with this, our approach utilizes the Web to obtain the collective contexts that engender extracting representative relations in social network. As

pointed in [168], we claim that relations should be defined not by local information, but rather by a global viewpoint of a network composed of individual relations.

As for modeling entity information from the Web, Conrad and Utt considered breaking up the corpus into what they called pseudo documents. They ran an entity recognizer through a corpus. All paragraphs containing a mention of an entity were collapsed into a single document called a pseudo document. They applied this to information visualization. We provide a more formal framework for the representation of these pseudo documents and extend the number of uses of this method of representation. Raghavan et al. explored the use of entity language models for tasks such as questio answering and clustering entities. To build the models, they recognized the named entities in the TREC-8 corpus and computed the probability distributions over words occurring within a certain distance of any instance labeled as Person of the canonical surface form of 162 famous people. In contrast, our approach for modeling entity information focus on the Web information. Therefore, we propose algorithms to model entity information from the Web using the information obitained from a search engine.

2.2 Semantic Web

With currently growing interest in the Semantic Web [119] and new standards for metadata description such as the Resource Description Framework (RDF) [132], metadata has gradually been becoming popular in the Web. Another recent trend in the Web is that the user is gradually coming to play a central role in Web contents. For example, in Weblog variety of contents is created by a user. And several Social Networking sites through which users can maintain an online network of friends or associates for social or business purposes have been launched recently. Therein, data about millions of people and their connections is publicly available on the Web.

With these recent Web trends, expressing semantics about people and their relationships has been gained interest. The Friend of a Friend (FOAF) project [169] is one of the Semantic Web's largest and most popular ontologies [123]. It is essentially a vocabulary for describing people and whom they know. The FOAF ontology isn't

the only one people use to publish social information on the Web. For example, it is reported that more than 360 RDF Schema or OWL classes defined with the local name "person" [1]. In fact, many vocabularies and frameworks for user semantics have being developed [133][122][175].

Users are begining to accept FOAF and its extensions as something of a standardized ontology for representing user semantics on the Semantic Web. However, as a major problem of the Semantic Web is in metadata annotation, metadata for users must also overcome the problem so that every user can easily annotate his or her data. The key clue to facilitate and accelerate metadata generation is to reuse much information which already has existed on the Web. In fact, while some FOAF files are from users who have authored their own data, others are from Web sites that publish data from their databases using the FOAF ontology. For example, imagine a researcher: that researcher's information can be found in an affiliation page, a conference page, an online paper, or in a Weblog.

Because recent studies have shown that social networks greatly contribute to ontology extraction [135], identifying underlying relations is important for ontology development. Currently, several studies are examining the use of relation extraction for ontology learning and population [156]. Although ontology learning and population share the common goal of facilitating ontology construction, they differ slightly. Whereas ontology learning mainly addresses extraction of taxonomic relations among concepts, the goal of ontology population is extraction of non-taxonomic relations among instances of concepts [157]. In our case, because the labels (non-taxonomic relations) of relations are assigned to pairs of entities of social networks (relation instances), our work can be regarded as a specific case of ontology population in the context of social networks.

Relation extraction for ontology population is typically an unsupervised approach. Because ontology population is usually intended to extract information about instances from large and heterogeneous sources such as the Web, a fully supervised approach that assumes numerous training instances is not feasible for large-scale exploitation, as pointed out in some precedent studies [158]. Therefore, several studies

[1] http://swoogle.umbc.edu

have exploited unsupervised or semi-supervised approaches. Particularly, the current approaches for relation extraction in ontology population are classifiable into two types: those that exploit certain patterns or structures, and those that rely on contextual features.

Pattern-based approaches [159, 160, 161] seek phrases or sentence structures that explicitly show relations between instances. However, most Web documents have a very heterogeneous structure, even within individual web pages. Therefore, the effectiveness of the pattern-based approach depends on the domain to which it is applied. Rather than exploiting patterns or structures, context-based approaches [162, 163, 164] assess contextual syntactic, semantic, and co-occurrence features. Several studies have employed contextual verb arguments to identify relations in text [162, 164], assuming that verbs express a relation between two ontology classes that specify a domain and range. Although verbs are relevant features to identify relations, we assume that syntactic and dependency analyses are applicable to text collections. Because the Web is highly heterogeneous and often unstructured, syntactic and dependency structures are not always available. For that reason, we employed a contextual model that uses a bag-of-words to assess context. Therefore, the method is applicable to any unstructured documents in the Web. As shown in our experiment, the simple context model performed well to extract descriptive relation labels without depending on any syntactic features in text.

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 [80, 81]. 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.

2.3 Social Network

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 [114]: Examples include viral marketing through social networks (also see [95]) and e-mail filtering based on social networks.

A social network is a social structure made of nodes which are generally individuals or organizations. It indicates the ways in which they are connected through various social familiarities ranging from casual acquintance to close familial bonds. Social network analysis is a technique in sociology, where a node is called an *actor* and an edge is called *tie*. From 1930's, social network analysis is applied to various kinds of network data. Recently, researchers such as D. Watts, Strogatz, and A. Newman develops the new network research area known as *complex network*.

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[2] and Orkut[3] 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 [86, 106] 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[4] and Yahoo! 360°[5] provide

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

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 [87]. 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 [103, 104]. 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 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 [72, 115, 105]. 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 [92]. In the mid-1990s, Kautz and Selman developed a social network extraction system from the Web, called *Referral Web* [92]. 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 [103][6]. 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.

Similarly, Y. Matsuo et al develops social network mining system from the Web [138, 168]. Their methods are similar to Flink and Referral Web, but they further developed to recognize the different types of relations and addressed the scalability. Their system, called *Polyphonet*, was 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) in order to promote the communication and collaboration among conference participants.

A. McCallum and his group [83, 74] 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. [90] develop a

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

system to extract names and also person-to-person relations from the Web. Faloutsos et al. [85] 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. [93] 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 [72]. Co-occurrence of terms in homepages can be a good indication to find communities, even obscure ones. Analyses of FOAF networks is a new research topic. To date, a couple of interesting studies have analyzed FOAF networks [123, 103]. Aleman-Meza et al. proposed the integration of two social networks: "knows" from FOAF documents and "co-author" from the DBLP bibliography [6]. They integrate the two networks by weighting each relationship to determine the degree of Conflict of Interest among scientific researchers.

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* [78], 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.

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 [94]. 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.

Chapter 3

Modeling Entity Information from Web

3.1 Entity Information

Large amounts of information about people, places and other entities are currently available in the Web. To extract and deduce information about these named entities has many practical applications. For example, if you are browsing the news site it would be interesting to be able to click on a name and get information about the entity associated with that name. A user might want to learn more about an entity by reading a summary of his or her identity, or might want specific questions about the entity answered. A user might also be interested in finding people related to the entity. It would also be interesting to find hidden attributes about an entity.

How is the entity represented so that information about entities can be used for pracitcal applications ? If we consider current information retrieval, it has been concerned with "document retrieval" rather than "entity retrieval" from a document collection, each represented as a bag of words. A user has a specific information need, and the system provides a list of documents that satisfy all or parts of that information need. Typically the list is presented in an order of decreasing relevance, where relevance is determined by the system. Often it is the user's job to connect the pieces of information together in order to satisfy a precise information need. On the other hand, there is increasing interest in more structured data in the Web. The Semantic Web relies on structured data and organized with rich ontologies. From this structured representation, a user can retrieve the data accoding his or her specific information requirement. However, one major issue of the Semantic Web approach is that much data in the Web is currently provided in the form of free text, lacking this structure.

In this thesis, we explore a middle ground between bag-of-words document retrieval and highly structured Semantic Web approach. The basic idea is that an entity can be represented as a weighted distribution of words that are likely to be used to describe the entity. Our hypothesis is that the highly weighted words will provide a useful representation of an entity. According to this basic assumption, we create a document-style representation of an entity using the contextual language around an entity in the Web. As processing documents, we can leverage the apply

Algorithm 3.2.1: ConstructEntityModel(e)

comment: Given entity e, return its entity model EM(e)

Query $q \leftarrow DisambiguateEntity(e)$
Document set $D \leftarrow WebSearch(q, n)$
for each document d in D
    $S \leftarrow Snippet(e, d, m)$
    for each snippet s in S
        Term $w \leftarrow TermExtraction(s)$
        Add term w into term set W
for each term w in W
    Assigning a entity type to term w using $NamedEntity(w)$
    Assigning a weight to term w using $WeightFunction(w)$
$NormalizeEntityModel(EM(e))$
return $(EM(e))$

Figure 3.1: Constructing Entity Model

the representation of an entity to several methods: extracting keywords about entities (Chapter 4), classifying entity relations and describing the semantics of the entity relations (Chapter 5), and extracting social network among entities (Chapter 6).

3.2 Constructing Entity Information Model

We represent an entity as a weighted distribution of words that are likely to be used to describe the named entity. We call the entity representation Entity Model (EM). For example, an entity model for "Shinzo Abe" would have "prime minister", "liberal democratic party", "cabinet", and other such words with high weight. It would also include names of strongly associated people (e.g., Junichiro Koizumi), places (Japan), and so on.

Constructing the entity model from the Web is outlined in Figure 3.1. We construct a model for an entity e as follows. First we create a search engine query from e. We collect the top n documents from a search result. For each document, we

Algorithm 3.2.2: CONSTRUCTENTITYTUPLEMODEL$(e1, e2)$

comment: Given entity pair e1 and e2, return entity tuple model ETM(e1,e2)

Query $q1 \leftarrow DisambiguateEntity(e1)$
Query $q2 \leftarrow DisambiguateEntity(e2)$
Document set $D \leftarrow WebSearch(q1\ and\ q2, n)$
for each document d in D
    $S \leftarrow Snippet(e1\ and\ e2, d, m)$
    for each snippet s in S
        Term $w \leftarrow TermExtraction(s)$
        Add term w into term set W
for each term w in W
    Assigning a entity type to term w using $NamedEntity(w)$
    Assigning a weight to term w using $WeightFunction(w)$
$NormalizeEntityModel(ETM(e1, e2))$
return $(ETM(e1, e2))$

Figure 3.2: Constructing Entity Tuple Model

find text contexts (we call these fragments snippets) that include e. The length of each snippet is defined with a text context spanning the m words to the right and to the left of e. For each snippet, we extract a term and add into a term list. After processing all snippets, we calculate a weight for each term in the term list. We also use a named entity extraction to provide an entity type (e.g., person, location, organization) for the term. Finally we obtain the Enitity Model (EM) for an entity e as a weighted distribution of terms. Each term is assigned a weight and an entity type (if the term is an entity). The entity model can be extended to higher-order features such as syntax. However, as we will show experimentally, our initial simple model is sufficient to make useful applications.

Using the same idea with a single entity model, we can also construct a Entity Tuple Model (ETM) that is a model of an entity pair. Constructing the entity tuple model from the Web is outlined in Figure 3.2. In case of an entity tuple model, we search for occurrences of the entity tuple in the text. If the number of intervating

words between the entities is less than a certain number, then we add up to the intervating words, up to the specified number of words to the right of the leftmost entity, and up to the specified number of words to the left of the rightmost entity to a snippet.

We define following functions for constracting entity model.

- *DisambiguateEntity*: Given an entity, it returns a search query including the entity and additional words to disambiguate the entity with other same-name entities.
- *WebSearch*: Given a search query, it returns top k documents that are retrieved by the query.
- *Snippet*: Given a text, an entity, the number of words to be included in a snippet , it returns text contexts (snippets) that include the entity.
- *TermExtraction*: Given a snippet, it returns terms in the snippet.
- *NamedEntity*: Given a term, it returns an entity type of the term.
- *WeightFunction*: Given a term and an entity, it returns a weight of the term in relation to the entity.

Below, we explain in details of above mentioned functions.

Entity Name Disambiguation

More than one person entity might have the same name. Such namesakes cause problems when constructing entity model. Several studies have addressed personal name disambiguation on the Web [74, 88, 97, 98]. In addition, the natural language community has specifically addressed name disambiguation as a class of word sense disambiguation [116, 99].

Bekkerman and McCallum uses probabilistic models for the Web appearance disambiguation problem [74]: 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 [96]. 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 constructing entity model, a relevant keyphrase to identify a person is useful because it can be added to a query. For example, an affiliation (a name of organization one belongs to) together with a name could be used to disambiguate namesakes. Given an entity, the *Disambiguate* entity function adds a couple of words that distinguish the entity from others. To this purpose, we cluster 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. For more details, please referer to [75].

Named Entity Extraction

Several studies in named entity recognition task have addressed identifying the type of named entities (such as people, locations, and organization). The named entity recognition task comprised three entity identification and labeling subtasks: ENAMEX (proper names and acronyms designating persons, locations, and organizations), TIMEX (absolute temporal terms) and NUMEX (numeric expressions, monetary expressions, and percentages). Several tools for identifing the type of named entities have been developed [177]. We combine these existing tools for the *NamedEntity* function to assign an entity type of a term.

Term Extraction

When extracting a term from a snippet, we have to deal with a compound noun that is made up of two or more words. To identify a compound noun, we use web counts information. Keller and Lapata investigated the validity of web counts for a range of predicate-argument bigrams (verb-object, adjective-noun, and noun-noun bigrams) [178]. They presented a simple method for retrieving bigram counts from the web by querying a search engine and showed that web-based frequencies can be a viable

alternative to bigram frequencies obtained from smaller corpora or recreated using smoothing. Following Keller and Lapata, we obtained web counts for n-grams using a simple heuristic based on queries to the search engine. In this approach, the web count for a given n-gram is simply the number of hits (pages) returned by the search engine for the queries generated for this n-gram. Using n-gram obtained from the search engine, we detect a compound noun.

Term Weighting

Our entity model is represented as a weighted distribution of words. Because our assumption is that high weighted words would provide a useful representation of an entity, the weighting function should evaluate relevancy that a word would represent an entity. If we regard an entity as a document, the word weighting is similar to weighting terms representing a document, which have been studied in information retrieval area.

In information retrieval, the weight functions are expressed as a product of a local weight function and a global weight function [179]. The local weight function presents the weight of the term in a document. The global weight function is used to express the weight of the term across the entire document set. If we apply these functions to our task of weighting terms in an entity model from the Web, the local weight corresponds to the weight of the term in documents from the search results of an entity and the global weight corresponds to the weight of the term across the entire Web documents.

We define two local weight functions: term frequency (tf_{web}) and logarithm ($logtf_{web}$). $tf_{web}(w)$ is defined as the term frequeny of term w in Document set D from the search results of an entity e. The term frequency in logarithmic scale is used to diminish the large numbers as follows:

$$logtf_{web}(w) = log(tf_{web}(w)) + 1).$$

As global weight functions, we define the following two functions: idf_{web} and

$entropy_{web}$. The global weight function idf_{web} is defined by:

$$idf_{web} = 1 + log(N_{web}/Hit(w)),$$

where N_{web} is the number of documents indexed by the search engine. In our case, we set $N = 10^{10}$ according to the number of indexed pages reported by Google. $Hit(w)$ is the hit counts retrieved by a query w.

The global weight function $entropy_{web}$ is defined by:

$$entropy_{web} = 1 - H(d|w)/H(d),$$

where $H(d)$ is the entropy of the disbribution (uniform) of the documents and $H(d|w)$ is the entropy of the conditional distribution given that the term w appeared.

Combining two local weight functions and two global weight functions, we have 4 weight functions. For example, using tf_{web} and idf_{web}, we obtain a $tf \cdot idf$ weighting that is often used as a weighting function in document retrieval.

Term Similarity

The similarity between an entity and a term can be also used as a weighting that express the relevancy of the term in relation to the entity. Many different methods have been proposed to measure the strength of word similarity [180] [68]. We use several word similarity measures as weighting funcitions as follows:

$$Dice_{web} = \frac{2 * Hit(e \cap w)}{Hit(e) + Hit(w)},$$

$$Overlap_{web} = \frac{Hit(e \cap w)}{min(Hit(e), Hit(w))},$$

$$Jaccard_{web} = \frac{Hit(e \cap w)}{Hit(e) + Hit(w) - Hit(e \cap w)},$$

$$PMI_{web} = \frac{Hit(e \cap w) * N_{web}}{Hit(e) * Hit(w)},$$

where e is an entity, w is a term, and Hit is the hit counts from the search engine.

If the probability distributions of an entity and a term are available, the following similarity functions can be used as weighing functions.

$$KL - divergence(p||q) = \Sigma plog\frac{p}{q},$$

$$Jensen - Shannon(p,q) = \frac{D(p||avg(p,q)) + D(q||avg(p,q))}{2},$$

$$SkewDivergence(p,q) = D(q||\alpha p + (1-\alpha)q),$$

$$Euclidean(p,q) = \sqrt{\Sigma(p-q)^2},$$

$$L1(p,q) = Sigma|p-q|,$$

$$Cosine(p,q) = \frac{\Sigma pq}{\sqrt{\Sigma p^2}\sqrt{\Sigma q^2}},$$

where p and q are probability distributions. The probability ditribution is calculated by using same algorithm of entity model construction and simply counting the term frequencies in all snippets.

3.3 Application of Entity Model

Our model is completely unstructured and based only on the text in the Web. In addition we do not employ any deep natural language processing beyond simple techniques nor do we use a knowledge base to improve our representation. Therefore, we expect that it can be ported to new domains with little difficulty. In particular our modeling approach provides an interesting new way to represent an entity, and it has broad applicability.

- Extracting keywords and populating Metadata of an entity
  Entity model could be used to extract the keywords of the entity. The keywords could be used for searching for an entity and answering to questions of an entity. Further, the keywords could be used to populate the Metadata together with the Semantic Web technologies.

- Classifying an entity into various categories
  Entity model could be used to group entities into classes.

- Linking entities that are similar and finding descriptions of why they are similar
  Entity model could be used to find links between entities and to provide meaningful descriptions of how two entities are related.

We demonstrate these applications of our entity model in following chapters: extracting keywords about entities (Chapter 4), classifying entity relations and describing the semantics of the entity relations (Chapter 5), and extracting social network among entities (Chapter 6).

Chapter 4

Entity Information Extraction from Web

In this chapter, we propose a method of extracting entity information in form of keyword from the Web. The proposed method is based on the statistical features of word co-occurrence that are obtained from search engine. The basic idea is a following: if a word co-occurs with an entity in many Web pages, the word might be a relevant keyword about the entity. Importantly, our method extracts relevant keywords depending on the context of the entity. Our evaluation shows better performance to $tfidf$-based keyword extraction. The keywords could be used to populate Metadata for the Semantic Web.

4.1 Introduction

With currently growing interest in the Semantic Web [119] and new standards for metadata description such as the Resource Description Framework (RDF) [132], metadata has gradually been becoming popular in the Web. Another recent trend in the Web is that the user is gradually coming to play a central role in Web contents. For example, in Weblog variety of contents is created by a user. And several Social Networking sites through which users can maintain an online network of friends or associates for social or business purposes have been launched recently. Therein, data about millions of people and their connections is publicly available on the Web.

With these recent Web trends, expressing semantics about people and their relationships has been gained interest. The Friend of a Friend (FOAF) project [169] is one of the Semantic Web's largest and most popular ontologies [123]. It is essentially a vocabulary for describing people and whom they know. The FOAF ontology isn't the only one people use to publish social information on the Web. For example, it is reported that more than 360 RDF Schema or OWL classes defined with the local name "person" [1]. In fact, many vocabularies and frameworks for user semantics have being developed [133][122][175].

Users are begining to accept FOAF and its extensions as something of a standardized ontology for representing user semantics on the Semantic Web. However, as a major problem of the Semantic Web is in metadata annotation, metadata for

[1] http://swoogle.umbc.edu

users must also overcome the problem so that every user can easily annotate his or her data. The key clue to facilitate and accelerate metadata generation is to reuse much information which already has existed on the Web. In fact, while some FOAF files are from users who have authored their own data, others are from Web sites that publish data from their databases using the FOAF ontology. For example, imagine a researcher: that researcher's information can be found in an affiliation page, a conference page, an online paper, or in a Weblog.

One of our research goals is to find information about person entity which already have been on the Web, and apply Semantic Web technologies to them. Therein, question is how we can find user's relevant information. In this chapter, we propose a novel keyword extraction method to extract entity information from the Web. The proposed method is based on the statistical feature of word co-occurrence. The basic idea is a following: if a word co-occurs with a person's name in many Web pages, the word might be a relevant keyword about his or her information. Importantly, our method extracts relevant keywords depending on the context of a person.

The remainder of this chapter is organized as follows: section 2 describes the proposed keyword extraction method. In section 3, we evaluate the method. In section 4, we discuss the limitation and application of our method in the Semantic Web. In section 5, we compare our method with related works. Finally, we conclude this chapter in section 6.

4.2 Keyword Extraction

4.2.1 Basic Idea

The simple approach to find someone's keyword is to use word co-occurrence information. Here, we define co-occurrence of two words as word appearance in the same Web page. If two words co-occur in many pages, it is assumed that those two have a strong relation. The co-occurrence information is acquired by the number of retrieved documents of a search engine result. For example the search result of a query "Alfred Kobsa and User Modeling" returns about 3100 documents while about 450

documents for a query "Alfred Kobsa and Software engineering". In this manner, we can guess that "User Modeling" is more relevant to "Alfred Kobsa" than "Software engineering". Our first hypothesis that:

Hypothesis1: The word that co-occurs with a person's name in many Web pages could be his or her keyword.

Although we can find many Web pages that contain a person's name, each page may contain personal information in different contexts. For example, imagine that one person who is both a researcher and an artist, we can expect that his name may appear not only in academic-related pages, but also in other pages related to his art activities. Even among his academic-related pages, there might be different pages depending on his acquaintances, affiliations, and projects. In this way, different Web pages reflect different contexts of a person. Here, we introduce the notion of a context to extract the keyword that captures the context of a person. We define a context word as word that describes someone's context. For example, "Art" and "Research" can be respectively context words for his art activities and research activities. Our second hypothesis that:

Hypothesis2: The word that co-occurs with a context word in many Web pages could be the keyword in the context.

4.2.2 Scoring Keywords based on Word Co-occurrence

Figure shows the algorihtm of the proposed keyword extraction and Fig. 4.2 shows procedures of the proposed keyword extraction. The proposed method has two main steps: (1) First step is to extract words that co-occur with a person's name in Web pages. (2) Second step is to give a score to each word using the degree of word co-occurrence in Web pages.

First, in order to extract words that co-occur with a person's name, we put his or her full name to a search engine As a search engine, we used Google [2] which currently addresses data from more than 8 billion Web pages. From the search result,

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

Algorithm 4.2.1: KEYWORDEXTRACTION(e)

comment: Given entity e and context c, return its keywords Keywords(e)

Query $q \leftarrow DisambiguateEntity(e)$
Document set $D \leftarrow WebSearch(q, n)$
for each document d in D
    $S \leftarrow Snippet(e, d, m)$
    for each snippet s in S
        Term $w \leftarrow TermExtraction(s)$
        Add term w into term set W
for each term w in W
    Assigning a entity type to term w using $NamedEntity(w)$
    Assigning a weight to term w using $Jaccard_{web}(e, w)$
    Assigning a weight in relation to context c using $Jaccard_{web}(c, w)$
    Score $s(w) \leftarrow score(Jaccard_{web}(e, w), Jaccard_{web}(c, w))$
$SortTerm(Keywords(e))$
return $(Keywords(e))$

Figure 4.1: Algorithm of keyword extraction

we used the top 10 html files as initial documents. The initial documents are pre-processed with html-tag deletion and part-of-speech (POS) tagging . Then, using the term extraction tool, Termex [3], we extract terms from pre-processed html files. Termex extracts terms from POS data based on statistical information of conjunctions between parts of speech. It can also extract nominal phrases that include more than two nouns such as "User Modeling". After the whole procedure of extraction, we extract about 1000 terms per person.

Based on the previous basic idea, the relevant keyword for a person is chosen based on word co-occurrence information. As a measure of co-occurrence, we use Jaccard coefficient that captures the degree of co-occurrence of two terms by their mutual degree of overlap. Jaccard coefficient is often used to evaluate tie strength between two objects [139]. Assume we are to measure the relevance of name n and

[3] http://gensen.dl.itc.u-tokyo.ac.jp/win.html

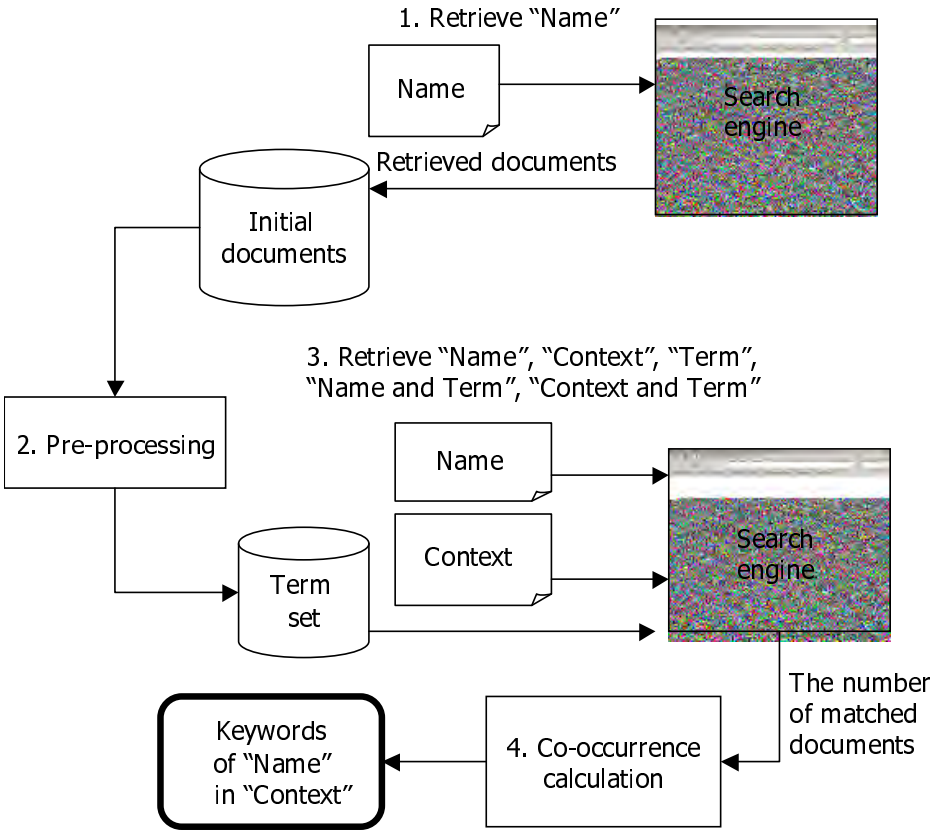

Figure 4.2: Procedure of keyword extraction

term w. We first put a query, "n and w", to a search engine and obtain the number of retrieved documents that is denoted by $Hit(N \cap W)$. Therein, N denotes a Web page set that includes n and W denotes a Web page set that includes w. We continuously apply a query, "n" and "w", and obtain the number of retrieved documents for each, $Hit(N)$ and $Hit(W)$. Then, the relevance between name n and term w, denoted by $Jaccard_{web}(n, w)$, is approximated by the following Jaccard coefficient.

$$Jaccard_{web}(n, w) = \frac{Hit(N \cap W)}{Hit(N \cup W)} = \frac{Hit(N \cap W)}{Hit(N) + Hit(W) - Hit(N \cap W)}$$

To extract the keyword in relation to a certain context, we need to estimate the relevance between the term and the context. If we replace the name n with the

context c in the relevance, $Jaccard_{web}(n, w)$, we can obtain the relevance between context c and term w, $Jaccard_{web}(c, w)$, in the same manner. Then, the relevance of person n and term w in the context c, denoted by $Score(n, c, w)$, is calculated as the following.

$$Score(n, c, w) = Jaccard_{web}(n, w) + \alpha Jaccard_{web}(c, w)$$

Therein, α denotes the relevance between the person and the context. We define threshold k for $Jaccard_{web}(n, c)$ to exclude terms that are not relevant for a person, but that have strong relation to the context. α and k are currently decided based on a heuristic method [4]. The term w with the higher $Score(n, c, w)$ is considered to be a more relevant keyword for person n in context c.

If we consider the relation between two persons in terms of their contexts, one person can be regarded as a part of the context of another person. Hence, we can apply the previous formula to keyword extraction of the relation between persons as follows:

$$RScore(n1, n2, c, w) = Score(n1, n2, w) + \beta\ Jaccard_{web}(c, w)$$

Therein, $n1$ and $n2$ denote each person's names in the relation. Context c can be considered in the relation between persons. β is the parameter of relevance between the persons and the context. This formula shows the term relevance of the relation between person $n1$ and $n2$ in the context c.

As an example of extracted keywords, Table 4.1 shows higher-ranked keywords of "Mitsuru Ishizuka". Each column in the table shows higher-ranked keywords based on $tfidf$, co-occurrence without the context, co-occurrence with the context "Artificial Intelligence", respectively, from the left column. Table 4.2 shows higher-ranked keywords with the context "University". Note that depending on the context word, context-related words (in bold type) come to appear in higher-ranked keywords. The

[4]For keywords in Table 4.1-4.3, we used as $\alpha = avg(Jaccard_{web}(n, w))/(3 * avg(Jaccard_{web}(c, w)))$, $k = 0.001$

Table 4.1: Higher-ranked keywords of "Mitsuru Ishizuka" using $tfidf$ and co-occurrence based method

$tfidf$	Co-Occurrence (without the context)	Co-Occurrence (with the context "Artificial Intelligence")
University of Tokyo	Yutaka Matsuo	**AI society**
University	Hiroshi Dohi	Yutaka Matsuo
JAVA application	Character Agent	**Natural Language**
Character	Koichi Hashida	Koichi Hashida
Scenario Emergence	Life-like Interface	Hiroshi Dohi
Research Institute	Naoaki Okazaki	Character Agent
Electronics	University of Tokyo	Life-Like Interface
Microsoft	Life-like Agent	Naoaki Okazaki
Iba laboratory	Hypothetical Reasoning	University of Tokyo
Yukio Osawa	Sadao Kurohashi	Life-like Agent
Program Committee	Life-like Internface	**AI journal**

order of higher-ranked keywords also changes in relation to the context. As an example of the relation keywords, Table 4.3 shows higher-ranked keywords between "Mitsuru Ishizuka" and "Yutaka Matsuo".

4.3 Evaluation

To evaluate the proposed method and validate our hypotheses, we extracted keywords of 10 Artificial Intelligence researchers. For each subject, we showed keywords that are extracted from the Web by tf (term freqeucy), $tfidf$ (term frequency inverse document frequency), *co-occur* (co-occurrence without the context), and our method (co-occurrence with the context). $tfidf$ is a method widely used by many keyword extraction systems to score individual words within text documents in order to select concepts that accurately represent the content of the document. $tfidf$ score of a word can be calculated by looking at the number of times the word appears in a document and multiplying that number by the log of the total number of documents (corpora) divided by the number of documents that the word resides in. As corpora, we used 3981 html files which are collected from the search results of 567 Japanese AI

Table 4.2: Higher-ranked keywords of "Mitsuru Ishizuka" with the context "University"

Co-occurrence with the context "University"
Yutaka Matsuo
Graduate School of Engineering
Hiroshi Dohi
Character Agent
Life-Like Interface
Artificial Intelligence
University of Tokyo
Faculty of Engineering
Life-life agent

Table 4.3: Higher-ranked keywords of the relation between "Mitsuru Ishizuka" and "Yutaka Matsuo"

Co-occurrence with the context "Artificial Intelligence"
National Institute of Advanced–
–Industrial Science and Technology
Artificial Intelligence
Ishizuka Laboratory
Naoaki Okazaki
Hiroshi Dohi
Yukio Osawa
Koishi Hashida
Naohiro Matsumura

Table 4.4: Precision, Coverage, Context Precision for 6 subjects

Method	tf	$tfidf$	*co-occur*	**ours**
precision	0.13	0.18	0.60	**0.63**
coverage	0.20	0.24	0.48	**0.56**
context precision	0.05	0.04	0.15	**0.19**

researchers'name. The *idf* is defined by $log(D/df(w))+1$, where D is the number of all documents and $df(w)$ is the number of documents including word w. In *co-occur*, keywords were extracted based on only co-occurrence between the name and term. In our method, we used "Artificial Intelligence" as the context word .

Using each method we extracted and shuffled the higher-ranked 20 terms derived each method. Then, the subjects were asked following three instructions:

I1 Check terms that are relevant to your research activities.

I2 Choose five terms that are indispensable for your research activities.

I3 Check terms that are relevant to your research activities from the viewpoint of Artificial Intelligence.

Precision was calculated by the ratio of the checked terms to 20 terms derived by each method (I1). Coverage of each method was calculated by taking the ratio of the indispensable terms included in the 20 terms to all the indispensable terms (I2). It is desirable to have the indispensable term list beforehand. However, it is very demanding for subjects to provide a keyword list without seeing a term list. In our experiment, we allowed subjects to add any terms to the indispensable term list even if they were not derived by any of the methods. Context precision is an evaluation criterion to measure how well context-related keywords are extracted. It is calculated by the ratio of the checked terms to 20 terms derived by each method (I3). Results are shown in Table 4.4. Compared with tf and $tfidf$, co-occurrence based methods exceed both in precision and coverage. tf and $tfidf$ select terms that appear frequently in the document (although $tfidf$ considers frequencies in other documents). On the other hand, co-occurrence based methods extract keywords in relation to another term even

if they do not appear frequently. This leads to better performance of co-occurrence based methods. With regard to context precision, our method that considers the context performs better than other methods. This means that our method can extract keywords in relation to the context (in this case, "Artificial Intelligence") better than other methods.

4.4 Discussion

Name Disambiguation

One problem of retrieving a person's name in a search engine is the case of two or more people having the same full name. One way to alleviate this same-name problem is to add a person's affiliation to the query. However, this degrades the coverage of search results. In particular, this makes the search focus more on one's activity in relation to the affiliation. It also excludes other contexts. It is necessary to solve the same-name problem without losing various contexts of people.

Privacy

While it is easy to obtain the information about researchers, ordinary people hardly expose their information in the Web. For further improvement of the proposed method, we must analyze what information is available about who in the Web, and its reliability. In this regard, Weblog and Social Network sites where people write variety of information are noteworthy subjects for the future. On the other hand, we should take care not to intrude on a user's privacy even in information extracted from the Web. A person sometimes does not know that his or her information is extracted from the Web only by name. We must clarify the use of information only for useful services for a user.

Dependency on Search Engine

The more we use Web pages and select keywords from whole pages, the greater the number of queries must be posted to a search engine later. For that reason, we used

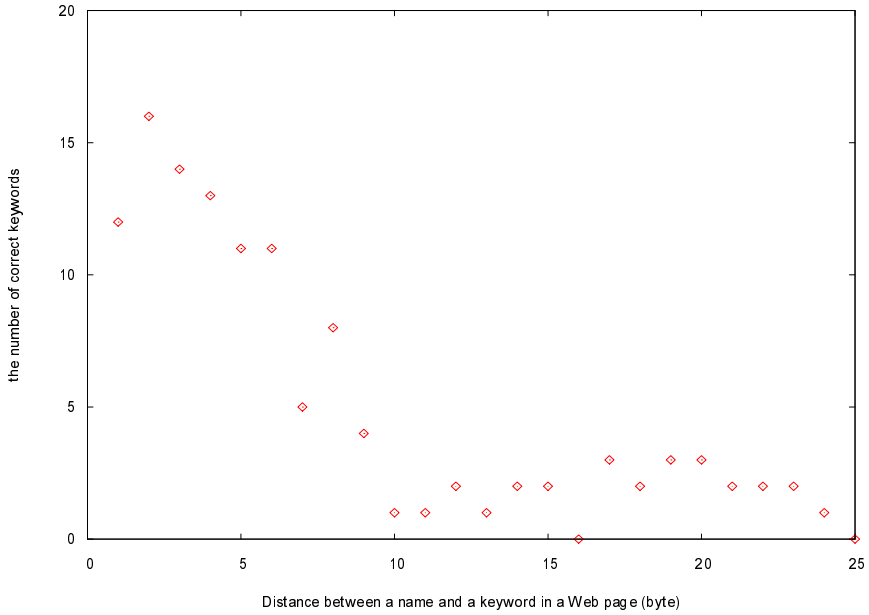

Figure 4.3: Distance between a name and a keyword vs. the number of correct keywords

the top 10 documents of search results to reduce the load of using a search engine. However, it is arguable how many documents of search results are to be used and whether the distance between a person's name and a keyword in a document is taken into account. Figure 4.3 shows the graph with the y-axis as the number of correct keywords that users chose in the experiment and the x-axis as the distance between a person'name and a keyword in a Web page. and Figure 4.4 shows the relation between the number of correct keywords and page-rank order of a search result. While most of keywords appear around a person's name and are contained in a higher-ranked page, some keywords are acquired indepedently of the distance and page-rank. We must examine these optimal parameters to extract adequate keywords.

Populating Medata from keywords

As shown Tables 4.1-4.3, keywords include various personal information such as person's name, organization, research project, and research interest. These are easily

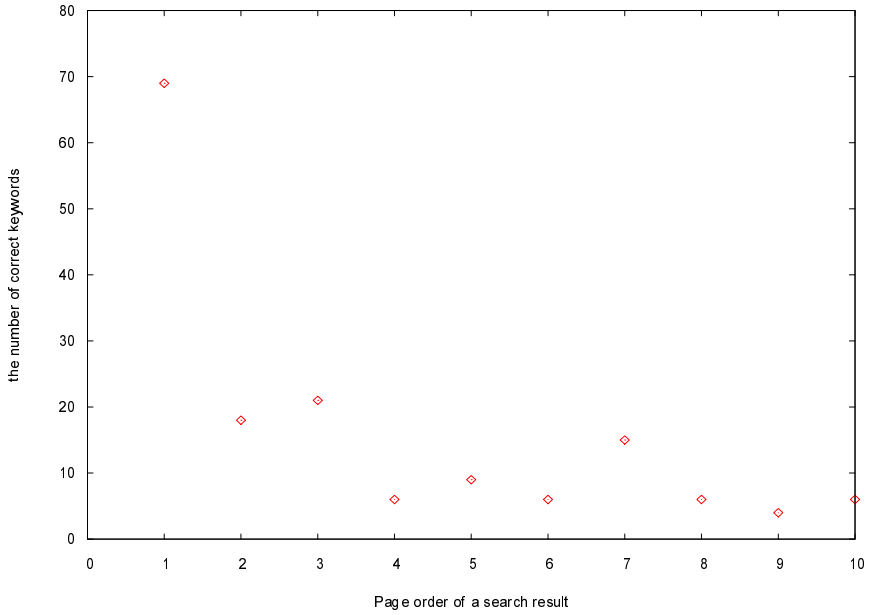

Figure 4.4: Page order of a search result vs. the number of correct keywords

incorporated in personal metadata, for example, FOAF properties:

foaf:knows, foaf:currentorganization, foaf:currentproject, foaf:interest

Currently, we are developing a method to automatically classify properties of keywords and generate personal metadata [127]. Figure 4.5 shows a FOAF file which is generated based on keywords. Using the keywords, we can facilitate the creation of a personal metadata file. We can also apply the metadata to partially annotate Web pages where keywords are extracted.

Once personal metadata or annotated Web pages is acquired, it can be very useful for a user profile in the Semantic Web. User adaptive system can use the profile for service such as recommendation and personalization. For example, the system might adapt to following user requests: Who knows this person? Who is involved in this project? Who knows this research topic well? Which pages include this person's information?

We addressed the importance of a person's context in keyword extraction. The context often defines kinds of properties. Currently, there is no FOAF vocabulary or

```
<foaf:Person>
<foaf:mbox rdf:resource=""/>
<foaf:name>Mitsuru Ishizuka</foaf:name>
<foaf:interest rdfs:label="Character agent" rdf:resource=""/>
<foaf:currentProject rdfs:label ="Life-like interface"
rdf:resource=""/>
<foaf:workplaceHomepage rdfs:label="University of Tokyo"
rdf:resource=""/>
<foaf:knows>
<foaf:Person>
<foaf:mbox rdf:resource=""/>
<foaf:name>Yutaka Matsuo</foaf:name>
......
```

Figure 4.5: An example of FOAF file based on extracted keywords.

its extensions to define a context. One way to introduce a personal context to those metadata frameworks is to prepare schema that corresponds to respective contexts. Regarding the expression of user semantics, we need further consideration to make it expressive and usable.

4.5 Related Work

Aiming at extracting keywords, our method is regarded as an IE (Information Extraction) method. Up to now, many IE methods have relied on predefined templates and linguistic rules or machine learning techniques to identify certain entities in text documents [131]. For example, some previous IE researches have addressed the extraction of personal information. In [118], the authors propose a method to extract artist information from Web pages, such as name and date of birth, and automatically generate his or her biography. In [124], they address the extraction of personal information such as name, project, publication in a specific department using unsupervised information extraction. These methods usually define properties, domains, or ontology beforehand. In contrast, we extract various information based on statistical word co-occurrence using only a name and a search engine without any predefined

restrictions.

Many keyword extraction methods for documents such as newspapers and scientific papers have been studied. In contrast to those documents, Web pages are too diverse and heterogeneous to apply the previous methods since they include free text and unstructured data, lack regular sentences. It is also difficult to apply probabilistic co-occurrence measures such as mutual information [121] and Log-Likelihood [125] since it is hard to estimate relevant population (the total number of Web pages and words) on the Web.

Some researches have focused on using a search engine to measure the stregth of relation between words [139, 176, 130]. They focus on extracting user's relationships or social network from the Web or the domain-specific terms. In our method, we can capture the various aspects of personal information from different Web pages using the notion of a context.

4.6 Conclusions

As users are gradually coming to play a central role in the Web contents, eliciting and representing personal information will increasingly be important in the user modeling research. In particular, with the currently growing trend toward the Semantic Web, expressing user semantics about people and the relations among them has been gained interest. This chapter proposed a novel method to extract entity information as keywords from the Web. Our evaluation showed better performance to $tfidf$-based keyword extraction.

Importantly, we use the Web as huge database and a search engine as its interface to obtain personal information in different contexts. While plenty of information is getting available on the Web, reusing and integrating online information of users will have significantly impact on personalization in the Semantic Web.

Chapter 5

Entities Relation Extraction from Web

In this chapter, we propose a method that automatically extracts descriptive labels of relations among entities automatically such as affiliations, roles, locations, part-whole, social relationships. Fundamentally, the method clusters similar entity pairs according to their collective contexts in Web documents. The descriptive labels for relations are obtained from results of clustering. The proposed method is entirely unsupervised and is easily incorporated with existing social network extraction methods. Our experiments conducted on entities in researcher social networks and political social networks achieved clustering with high precision and recall. The results showed that our method is able to extract appropriate relation labels to represent relations among entities in the social networks.

5.1 Introduction

Social networks have recently attracted considerable interest. For the Semantic Web, there is great potential to utilize social networks for myriad applications such as trust estimation [134], ontology construction [135], and end-user ontology [136].

Aiming at using social networks for AI and the Semantic Web, several studies have addressed extraction of social networks automatically from various sources of information. Mika developed a system for extraction, aggregation, and visualization of online social networks for a Semantic Web community, called Flink [137]. In that system, social networks are obtained using Web pages, e-mail messages, and publications. Using a similar approach, Matsuo et al. developed a system called Polyphonet [138, 168]. In line with those studies, numerous studies have explored automatic extraction of social networks from the Web [6]. We also address the extraction of social networks in Chapter 6.

Given social network extraction using the methods described above, the next step would be to explore underlying relations behind superficial connections in those networks. In the field of social network analysis, it has been shown that rich information about underlying social relationships engenders more sophisticated analysis [143]. However, most automatic methods to extract social networks merely provide a clue to the strength of relations. For example, a link in Flink [137] is only assigned the

strength of its relation. A user might wonder what kind of underlying relation exists behind the link.

One reason for the lack of information about underlying relations is that most automatic extraction methods [137, 6, 138] use a superficial approach (e.g. co-occurrence analysis) instead of profound assessment to determine the type of relation. Matsuo et al. defines four kinds of relations in a research community and classifies the extracted relations. They adopt a supervised machine learning method, which requires a large annotated corpus which costs great deal of time and effort to construct and administer. In addition, it is necessary to gather domain-specific knowledge a priori to define the extracted relations.

Our goal is to extract underlying relations among entities (e.g., person, location, company) from social networks (e.g., person-person, person-location network). Thereby, we are aiming at extracting descriptive labels of relations automatically such as affiliations, roles, locations, part-whole, social relationships. In this chapter, we propose a method that automatically extracts the labels that describe relations among entities in social networks. We obtain a local context in which two entities co-occur on the Web, and accumulate the context of the entity pair in different Web pages. Given the collective contexts of each entity pair, the key idea is clustering all entity pairs according to the similarity of their collective contexts. This clustering using collective contexts is based on our hypothesis that entity pairs in similar relations tend to occur in similar contexts. The representative terms in context can be regarded as representing a relationship. Therefore, the labels to describe the relations among entities are extracted from the clustering process result. As an exemplary scenario for our approach, we address two kinds of social network. a researcher social network and a political social network.

The proposed method is entirely unsupervised. For that reason, our method requires neither a priori definition of relations nor preparation of large annotated corpora. It also requires no instances of relations as initial seeds for weakly supervised learning. Our method uses context information that is obtained during extraction of social networks. Consequently, the proposed method contributes to

- incorporating into existing methods of social network extraction and enriching

the social network by adding relation labels.

In addition, as a Web mining approach our method contributes to

- extracting information from the Web and bootstrapping the Semantic Web by annotating relation information to social networks and Web contents.

The remainder of this chapter is structured as follows. Section 2 compares our approach to other ongoing relevant research in social network extraction, relation extraction, and ontology population. Section 3 describes basic ideas of our approach and detailed steps of the proposed method. Section 4 describes our experiment. Section 5 describes results and evaluation. We end our presentation with a discussion of future work, after which we provide concluding remarks in section 6.

5.2 Related Work

Aiming at extracting underlying relations in social networks from the Web, our method is related closely to existing extraction methods of social networks. Several studies have addressed extraction of social networks automatically from various sources of information such as the Web, e-mail, and contacts [6, 46, 137, 168]. While most approaches for social network extraction have focused on the strength of the relation, few studies have addressed automatic identification of underlying relations. Matsuo et al. employed a supervised machine learning method to classify four types of relations in a research community [138]. There have also been several important works that have examined supervised learning of relation extraction in the field of natural language processing and information extraction [151, 152, 153]. However, a supervised method requires large annotated corpora, which cost a great deal of time and effort. In addition, it is necessary to gather the domain specific knowledge a priori to define extracted relations.

Identifying underlying relations has been also addressed in ontology population [156]. Particularly, the current approaches for relation extraction in ontology population are classifiable into two types: those that exploit certain patterns or structures, and those that rely on contextual features. Pattern-based approaches [165]

seek phrases or sentence structures that explicitly show relations between instances. However, most Web documents have a very heterogeneous in structure, even within individual web pages. Therefore, the effectiveness of the pattern-based approach depends on the domain to which it is applied.

Rather than exploiting patterns or structures, context-based approaches [162, 164] assess contextual features such as syntactic, semantic, and co-occurrence. Several studies have employed contextual verb arguments to identify relations in text [162, 164], assuming that verbs express a relation between two ontology classes that specify a domain and range. Although verbs are relevant features to identify relations, it assumes that syntactic and dependency analyses are applicable to text collections. Because the Web is highly heterogeneous and often unstructured, syntactic and dependency structures are not always available. For that reason, we employed context model that uses a bag-of-words to assess context. Therefore, the method is applicable to any unstructured documents in the Web. As shown in our experiment, the simple context model performed well to extract descriptive relation labels without depending on any syntactic features in text. In the field of NLP, several studies have proposed relation extraction from a large language corpus using a bag-of-words of context [68, 33]. Our method can be considered as an application of relation extraction methods in NLP to social networks and a Web mining. We are aiming at easily incorporating into extraction methods of social network from the Web. Therefore, our method uses context information that is obtained during extraction of social networks. Consequently, it serves to enrich such networks by adding relation labels.

5.3 Method

5.3.1 Concept

In this chapter, as an exemplary scenario for our approach, we use two types of social network: a researcher network that is composed of researcher entities and a political social network that is composed of two types of entities: politicians and geo-political-entities. Figure 5.2 and 5.2 shows a political social network and a social network

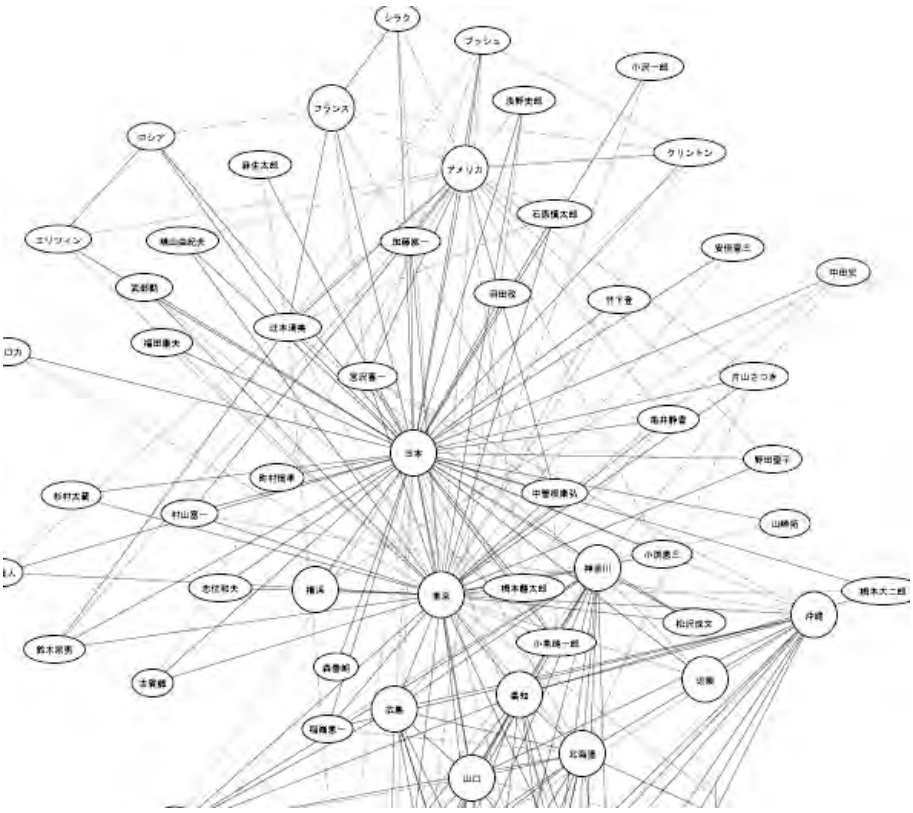

Figure 5.1: Political social network extracted from the Web: a circular node represents a location entity and a ellipse node represents a person entity. Each edge in the network implies that there is relation between entities.

of Japanese AI researchers respectively. The social networks were automatically extracted from the Web using the method proposed in Chapter 6. The method uses co-occurrence of entities on the Web to access the relation between entities.

Given entity pairs in the social network, our present goal is to extract labels to describe the relations of respective entity pairs (to discover relevant keyphrases that relate entities). The simple approach to extract the labels that are useful for describing relations in social networks is to analyze the surrounding local context in which entities of interest co-occur on the Web, and to seek clues to describe that relation. Local context is often used to identify entities or relations among entities in tasks of natural language processing or information extraction [145, 146, 147].

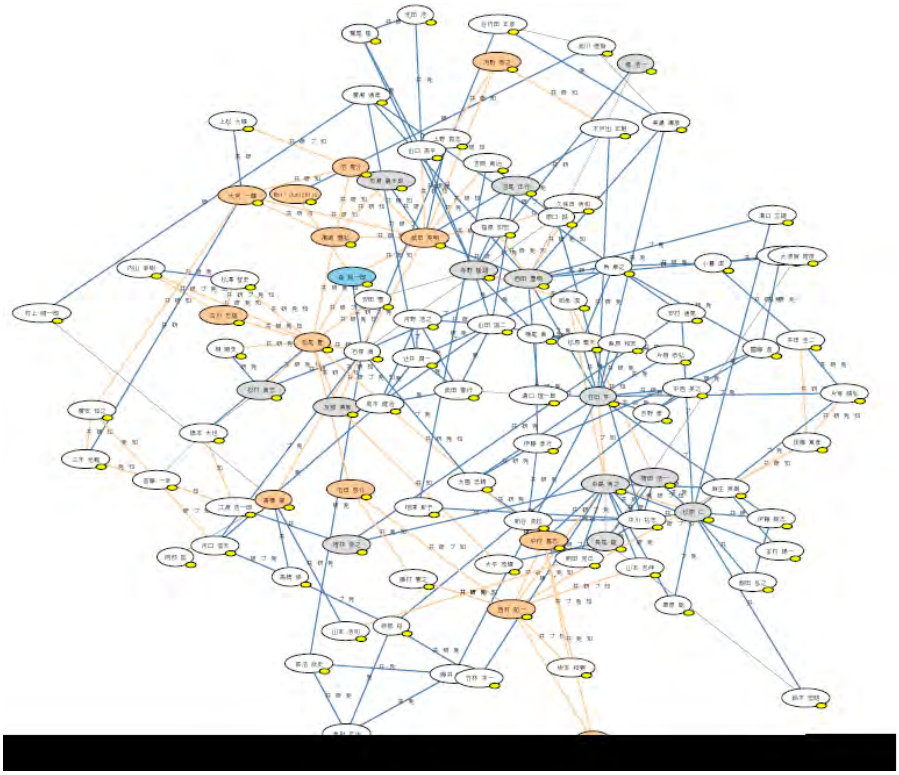

Figure 5.2: Social network of Japanese AI researchersextracted from the Web: a circular node represents a researcher entity. Each edge in the network implies that there is relation between researchers.

Table 5.1 shows keywords [1] that were extracted from local contexts of four entity pairs (Junichiro Koizumi-Japan, Yoshiro Mori-Japan, Junichiro Koizumi-Kanawaga, Yoshiro Mori-Ishizuka) using keyword extraction method [138]. The keywords were extracted from the collective local contexts where co-occurrence of each entity pair is found. For each entity pair, the local contexts from 100 Web pages were collected. The keywords are ordered according to TF-IDF-based scoring, which is a widely used method in many keyword extraction methods to score individual words within text documents to select concepts that accurately represent the documents' contents. The keywords scored by TF-IDF can be considered as a bag-of-words model to represent

[1] In our experiment, we mainly used Web pages in Japanese. Therefore, keywords in the table are translated from their original Japanese.

Table 5.1: Keywords obtained from each local context of four kinds of entities pairs: Junichiro Koizumi-Japan, Yoshiro Mori-Japan, Junichiro Mori-Kanagawa, and Yoshiro Mori-Ishikawa

(1) Junichiro Koizumi-Japan
pathology, Fujiwara, **prime minister**, Koizumi, Kobun-sha, politics, visit, page, products, cabinet,...

(2) Yoshiro Mori-Japan
rugby, **prime minister**, chairman, bid, minister, association, science, administration, director, soccer, Africa,...

(3) Junichiro Koizumi-Kanagawa
election, **prime minister**, Yokosuka, **candidate**, **congressional representative**, Saito, **Liberal Democratic Party**, Miura,..

(4) Yoshiro Mori-Ishikawa
Ichikawa, Yasuo, **prime minister**,**election**, **Liberal Democratic Party**, Okuda, **candidate**, komatsu, **congressinal representative**,...

the local context surrounding an entity pair.

If we examine the common keywords (shown in bold typeface in the table) shared by (1) and (2) or (3) and (4), we note that the keywords that describe the relations of each entity pair, such as "prime minister" and "candidate", are commonly shared [2]. In contrast, if we compare Koizumi's keywords (1) with another of his keywords (3), we find that different keywords appear because of their respective links to different locations: Japan and Kanagawa. (although both keywords are Koizumi's.)

Based on the observations described above, we hypothesize that if the local contexts of entity pairs in the Web are similar, then the entity pairs share a similar relation. Our hypothesis resembles previously tested hypotheses related to context [148, 147]: words are similar to the extent that their contextual representations are

[2]Junichiro Koizumi is the current Prime Minister of Japan and Yoshiro Mori is a former Prime Minister. Kanagawa is the prefecture where Koizumi was elected and Ishikawa is the prefecture where Mori was elected.

similar. According to that hypothesis, our method clusters entity pairs according to the similarity of their collective contexts. Then, the representative terms in a cluster are extracted as labels to describe the relations of each entity pair in the cluster, assuming that each cluster represents different relations and that the entity pair in a cluster is an instance of a certain relation. The key point of our method is that we determine the relation labels not by examining the local context of one single entity pair, but by the collective local contexts of all entity pairs of interest. In the following section, we explain the precise steps of our proposed method.

5.3.2 Procedure

Our method for extraction of relation labels in social networks includes the following steps.

1. Collect co-occurrence information and local context of an entity pair
2. Extract a social network that is composed of entity pairs.
3. Generate a context model of each entity pair.
4. Calculate context similarity between entity pairs.
5. Cluster entity pairs.
6. Select representative labels to describe relations from each cluster.

Figure 5.3 depicts the outline of our method. Figure **??** shows the algorithm of our method. Our method requires a list of entities (e.g., personal name, location name) to form a social network as the input; it then outputs the social network and a list of relation labels for each entity pair. Although collection of a list of entities is beyond the scope of this chapter, one might use named entity recognition to identify entities and thereby generate a list of entities of interest.

As shown in Fig. 5.3, our method (upper box) can be processed in parallel with existing methods of social network extraction (bottom box). The first step is to collect co-occurrence and local contexts of each entity pair from the Web. Many

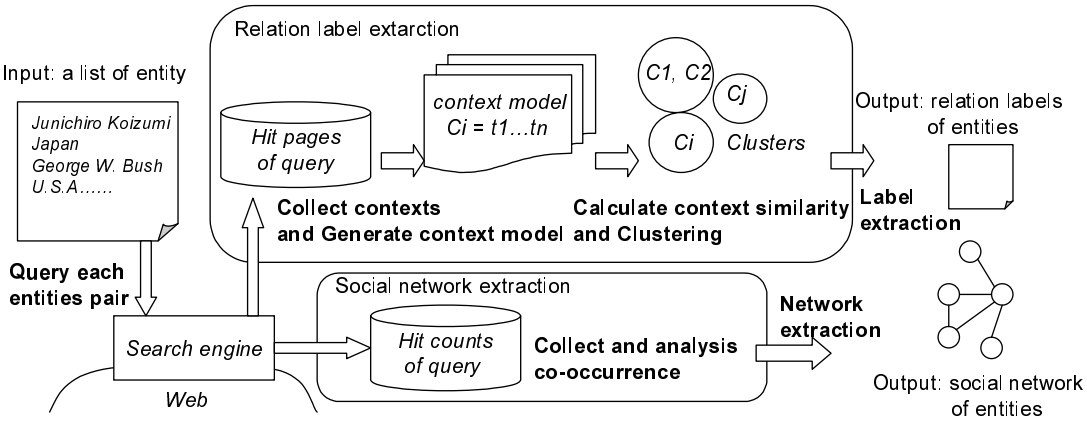

Figure 5.3: Outline of the proposed method

existing methods of social network extraction use a search engine and its query hit counts to obtain co-occurrence information of entities from the Web. In line with such methods, we use Google [3] to collect co-occurrence information and generate a social network, as shown in Fig. 5.2 and 5.2.

Using co-occurrence information, we also collect local contexts in which elements of an entity pair of interest co-occur within a certain contextual distance of one another within the text of a Web page. For this, we downloaded the top 100 web pages included in the search result of corresponding search query to each entity pair (in our example of a politician and location name, the query is "Junichiro Koizumi AND Japan"). This can be accomplished in the process of collecting co-occurrence information, which uses search query hit counts.

5.3.3 Context Model and Similarity Calculation

For each entity pair, we accumulate the context terms surrounding it; thereby, we obtain the contexts of all entity pairs. As the next step, to calculate the similarity between collective contexts of each entity pair, we require a certain model that represents the collected context. In our method, we propose a context model that represents the context using a bag-of-words and a word vector [149]. We define the

[3]http://www.google.com

Algorithm 5.3.1: RELATIONEXTRACTION(e)

comment: Given a list of entity pairs EP in social network,

comment: return its relation labels RL(EP)

for each entity pair $e1$ *and* $e2$ in EP
Context model $C(e1, e2) \leftarrow ConstructContextModel(e1, e2)$
Add context model $C(e1, e2)$ into context model set Cs
for each pair C_1 *and* C_2 in Cs
    Calculating context similarity using $sim(C_1, C_2)$
$Clustering(EP)$
$LabelExtraction(EP)$
return $(RL(EP))$

Figure 5.4: Algorithm of the propsed method

context model as a vector of terms that are likely to be used to describe the context of an entity pair (e.g., the keywords list shown in Table 5.1 can be considered as an example of the context model.).

Figure 5.5 shows the algorithm for constructing the context model of an entity pair. A context model $C_{i,j}$ of an entity pair (e_i, e_j) is defined as the set of N terms $t_1, ..., t_N$ that are extracted from the context of an entity pair as $C_{i,j}(n, m) = t_1, ..., t_N$, where both n and m are parameters of the context window size, which defines the number of terms to be included in the context. In addition, m is the number of intervening terms between e_i and e_j; n is the number of words to the left and right of either entity.

Each term t_i in the context model $C_{i,j}(n, m)$ of an entity pair (e_i, e_j) is assigned a feature weight according to TF-IDF-based scoring defined as

$$tf(t_i) \cdot idf(t_i).$$

Therein, $tf(t_i)$ is defined by the term frequency of term t_i in all the contexts of the entity pair (e_i, e_j). Furthermore, $idf(t_i)$ is defined as $log(|C|/df(t_i))+1$, where $|C|$ is

Algorithm 5.3.2: CONSTRUCTCONTEXTMODEL($e1, e2$)

comment: Given entity pair e1 and e2, return entity tuple model C(e1,e2)

Query $q1 \leftarrow DisambiguateEntity(e1)$
Query $q2 \leftarrow DisambiguateEntity(e2)$
Document set $D \leftarrow WebSearch(q1\ and\ q2, 100)$
for each document d in D
    $S \leftarrow Snippet(e1\ and\ e2, d, m, n)$
    for each snippet s in S
        Term $w \leftarrow TermExtraction(s)$
        Add term w into term set W
for each term w in W
    Assigning a weight to term w using $tf(w) \cdot idf(w)$
$NormalizeContextModel(C(e1, e2))$
return $(C(e1, e2))$

Figure 5.5: Algorithm of constructing context model

the number of all context models and $df(t_i)$ is the number of context models including term t_i. With the weighted context model, we calculate the similarity $sim(C_{i,j}, C'_{i,j})$ between context models according to the cosine similarity as follows:

$$sim(C_{i,j}, C'_{i,j}) = C_{i,j}C'_{i,j}/(|C_{i,j}||C'_{i,j}|).$$

In our exploratory experiment we tried probability distribution-based scoring and several similarities such as L1 norm, Jensen-Shannon and Skew divergence [146]. According to that results, TFIDF-based cosine similarity performs well.

5.3.4 Clustering and Label Selection

Calculating the similarity between the context models of entity pairs, we cluster all entity pairs according to their similarity. This is based on our hypothesis: the local contexts of entity pairs in the Web are similar, the entity pairs share a similar relation.

In our clustering process, we do not know in advance what kinds of relation pertain

and therefore how many clusters we should make. Therefore, we employ hierarchical agglomerative clustering. Several clustering methods exist for hierarchical clustering. According to our exploratory experiment, complete linkage performs well because it is conservative in producing clusters and does not tend to generate a biased large cluster. In complete linkage, the similarity between the clusters CL_1, CL_2 is evaluated by considering the two most dissimilar elements as follows.

$$min_{C_{i,j} \in CL_1, C'_{i,j} \in CL_2} sim(C_{i,j}, C'_{i,j}).$$

The clustering process terminates when cluster quality drops below a predefined threshold. The cluster quality is evaluated according to two measures [150]: the respective degree of similarity of entity pairs within clusters and among clusters. After the clustering process terminates and creates a certain number of clusters, we extract the terms from a cluster as labels to describe the relations of each entity pair in the cluster. This is based on our assumption that each cluster represents a different relation and each entity pair in a cluster is an instance of similar relation. The term relevancy, as a cluster label, is evaluated according to a TFIDF-based measure in the same manner as weighting the terms in a context model. However, in this process, the term frequency is determined for all contexts of a cluster. The underlying idea is to extract terms that appear in the cluster, but which do not appear in other clusters. With a cluster CL's labels $l_1, ..., ln$ scored according to the term relevancy, an entity pair, e_i and e_j, that belongs to the CL can be regarded as holding the relations described by $l_1, ..., ln$.

5.4 Experiment

Using our proposed method, we extracted labels to describe relations of each entity pair in a social network. We chose 143 distinct entity pairs (pair of a politician and a geo-political entity) from a political social network and 421 entity pairs (pair of Japanese AI researchers) from a researcher network [168].

We created a context model of each entity pair using nouns and noun phrases from

part-of-speeches (POS) of surrounding entity pairs in a Web page. We exclude stop words, symbols and highly frequent words. For each entity pair, we download the top 100 web pages in the process of collecting co-occurrence information for extraction of social network. For the context size, we used two parameters, m and n, as explained in Sect. 5.3.3. As a baseline of the context size, we assigned 10 and 5, respectively, to m, n.

We used complete-linkage agglomerative clustering to cluster all entity pairs. Thereby, we created five distinct clusters for the political social network and twelve distinct clusters for the researcher network according to the predefined thresholds of two quality measures within the clusters and among the clusters explained in Sect. 5.3.4. To evaluate the clustering results and the extracted labels, two human subjects analyzed the context terms of each entity pair and manually assigned the relation labels (three or fewer possible labels for each). Then, a cluster label was chosen as the most frequent term among the manually assigned relation labels of entity pairs in the cluster. The manually assigned relation labels are used as ground truth in the subsequent evaluation stage.

In Table 5.2, the left column shows the label of each cluster. The right column shows the highly-scored terms[4] that are extracted automatically from each cluster, which can be considered as the labels to describe relations of each entity pair in the cluster. The terms are sorted by relevancy score.

5.5 Evaluation

We first evaluated the clustering results using a political social network. For each cluster cl, we counted the number of entity pairs $EP_{cl,correct}$ whose manually assigned relation labels include the label of cluster cl. We also counted the entity pairs $EP_{cl,total}$ in the cluster cl. Next, for each relation label l, we counted the number of entity pairs $EP_{l,correct}$ that have the relation label l whose cluster label is l. We also counted the entity pairs $EP_{l,total}$ that have the relation label l. Then, precision and recall of the

[4]Terms in the table are translated from their original Japanese.

Table 5.2: Cluster label (left) and automatically extracted relation labels from a cluster (right)

political social network (5 clusters)

1	mayor	mayor, citizen, hosting, president, affairs, officer, mutter, answer, city, conference
2	president	president, administration, world, Japan, economics, policy, war, principle, politics, Iraq
3	prime minister	prime minister, administration, politics, article, election, prime minister, government, peace
4	governor	prefectural governor, president, prefectural government, committee, Heisei, prefectural administration, mayor
5	congressional representative	congressional representative, election, Liberal Democratic Party, candidate, lower house, Democratic Party, proportional representation

researcher social network (6 representative clusters among 12 clusters)

1	co-authorship of conference paper	paper, author, conference venue, presentation, title, program
2	co-authorship of book	edit, book, publishing, programming, recommendation, co-author
3	co-edit of book	edit, revision, article, publishing, educational material, editor
4	collaborative project	representative person, contributor, minister, acceptance
5	co-authorship of journal paper	journal, Shogi, distribution, computer, information processing society of Japan
6	same affiliation	University of Tokyo, metropolitan, techonolgy, University, science

cluster were calculated as:

$$precision = \Sigma_{cl \in CL} \frac{EP_{cl,correct}}{EP_{cl,total}},\ recall = \Sigma_{l \in L} \frac{EP_{l,correct}}{EP_{l,total}}.$$

According to *precision* and *recall*, we evaluated clusters based on the F measure.

The graph depicted in Fig. 5.6 shows that the clustering results vary depending on the context size. Consequently, to find the optimal context size, we calculate the F-measure by changing two size parameters: m and n. Expanding the context size from the minimum, the F-measure takes an optimal value when m is around 30 and n is around 10 (Fig. 5.6 and Table 5.4) . We employed this optimal context size to

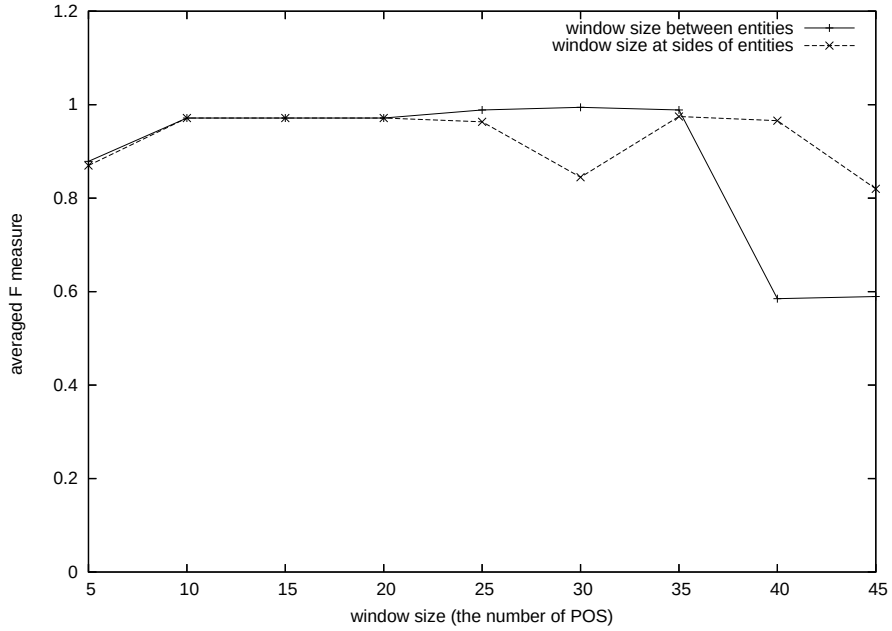

Figure 5.6: F measure of clustering results vs. Context window size

Table 5.3: Clustering performance in parameters of context window size

Context window size n, m	Precision	Recall	F-measure
$n = 10$, $m = 30$	0.992	0.995	0.994
$n = 5$, $m = 10$	0.88	0.85	0.86
All terms in a Web page	0.76	0.677	0.716

extract the relation labels in our experiment. After reaching the peak, the value of the F-measure decreases as the context size increases. The wider context window tends to include noise terms that are not appropriate to represent the context, thus rendering the similarity calculation between the contexts irrelevant. The optimal context size depends on the structural nature of language. Consequently, we must choose the context size carefully when applying our methods to a different language.

To evaluate the automatically extracted relation labels, we compared the cluster label (left column of Table 5.2) with the automatically extracted relation labels (right column of Table 5.2). For a political social network, we found that the relation

label that has the highest score is equal to the corresponding cluster's relation label. Precision of the clustering results in our experiment is quite high, as shown above. Therefore, we can say that each entity pair in a cluster is represented properly by the highest-scored relation labels. For a researcher social network, extracted relation labels are highly correlated with a manually assigned clustering label. Matsuo et al. defined four kinds of relations for a research social network: co-authorship, same affiliation, same project, and same conference [138]. We found that extracted clusters and relation labels are corresponding those relations.

5.6 Conclusions and Future Work

We propose a method that automatically extracts labels that describe relations between entities in social networks. The proposed method is entirely unsupervised and domain-independent; it is easily incorporated into existing extraction methods of social networks.

Recent important approaches of a Web mining toward the Semantic Web use the Web as a huge language corpus and combine with a search engine. The underlying concept of these methods is that it uses globally available Web data and structures to annotate local resources semantically to bootstrap the Semantic Web. In line with this, our approach utilizes the Web to obtain the collective contexts which engender extracting representative relations in social network. As pointed in [168], we claim that relations should be defined not by local information but rather by a global viewpoint of a network composed of individual relations.

Future studies will explore the possibilities of extending the proposed method to relations in other types of social networks. Enriching social networks by adding relation labels, our method might contribute to several social network applications such as finding experts and authorities, trust calculation, community-based ontology extraction, and end user ontology.

Chapter 6

Social Network Extraction from Web

In this chapter, we propose a method that automatically extracts social networks from the Web. The Web is currently a huge source of information for the relation between entities. Our method leverages co-occurrence information obtained from a search engine to extract a social network among entities. The basic idea is a following: if two entities co-occurs in many Web pages, they might have a relation. We evaluated several co-occurrence measures to find the robast measure for extracting a social network from the Web. Combining several information about entity and relation that are also extracted automatically from the Web, we develop a method for extracing a social network in the way that the social network is easy to understand and applicable for practical applications.

6.1 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 [114]: Examples include viral marketing through social networks (also see [95]) and e-mail filtering based on social networks.

A social network is a social structure made of nodes which are generally individuals or organizations. It indicates the ways in which they are connected through various social familiarities ranging from casual acquintance to close familial bonds. Social network analysis is a technique in sociology, where a node is called an *actor* and an edge is called *tie*. From 1930's, social network analysis is applied to various kinds of network data. Recently, researchers such as D. Watts, Strogatz, and A. Newman develops the new network research area known as *complex network*.

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 [86, 106] 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 [87]. 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 [103, 104]. 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 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.

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

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 [72, 115, 105]. 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 [92]. Several researchers have used that technique to extract social networks, as described in the next section.

In this chapter, we propose a method that automatically extracts social networks from the Web. The Web is currently a huge source of information for the relation between entities. Our method leverages co-occurrence information obtained from a search engine to extract a social network among entities. The basic idea is a following: if two entities co-occurs in many Web pages, they might have a relation. We evaluated several co-occurrence measures to find the robast measure for extracting a social network from the Web. Combining several information about entity and relation that are also extracted automatically from the Web, we develop a method for extracing a social network in the way that the social network is easy to understand and applicable for practical applications.

This chapter is organized as follows. The following section describes related studies and motivations. Section 3 addresses basic algorithms to obtain social networks from the Web. In Section we discuss some issues and future trends in the social network extraction. We describe the research search system as an application of social networks in Section 4. Then we conclude this chapter.

6.2 Related Work

In the mid-1990s, Kautz and Selman developed a social network extraction system from the Web, called *Referral Web* [92]. 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 [103][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.

Similarly, Y. Matsuo et al develops social network mining system from the Web [138, 168]. Their methods are similar to Flink and Referral Web, but they further developed to recognize the different types of relations and addressed the scalability. Their system, called *Polyphonet*, was operated at the 17th–21st Annual Conferences of the Japan Society of Artificial Intelligence (JSAI2003–2007) and at The International Conference on Ubiquitous Computing (UbiComp 2005) in order to promote the communication and collaboration among conference participants.

A. McCallum and his group [83, 74] 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. [90] develop a

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

system to extract names and also person-to-person relations from the Web. Faloutsos et al. [85] 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. [93] 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 [72]. Co-occurrence of terms in homepages can be a good indication to find communities, even obscure ones. Analyses of FOAF networks is a new research topic. To date, a couple of interesting studies have analyzed FOAF networks [123, 103]. Aleman-Meza et al. proposed the integration of two social networks: "knows" from FOAF documents and "co-author" from the DBLP bibliography [6]. They integrate the two networks by weighting each relationship to determine the degree of Conflict of Interest among scientific researchers.

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 [80, 81]. 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* [78], 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 [94]. 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.

6.3 Social Network Extraction from the Web

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 (Japan Society for Artificial Intelligence) community as an example, and show some results on extracting the community.

6.3.1 Node and Edge Extraction

Figure 6.1 shows the algorithm for extracting social networks. A social network is extracted through two steps. First we obtain nodes, then we add edges. Starting from a center node, we subsequently create new nodes by finding person names from the search results iteratively. Therefore, we can expand the network one node at a time. Several studies including the Referral Web and McCallum's study also have employed same approach to expand the network. In some studies, nodes in a social network are given, namely a list of persons is given beforehand.

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.

Algorithm 6.3.1: ExtractSocialNetwork(e)

comment: Given entity e, return its social network SN(e,V,E)

Query $q \leftarrow DisambiguateEntity(e)$
Document set $D \leftarrow WebSearch(q, n)$
for each document d in D
    $S \leftarrow Snippet(e, d, m)$
    for each snippet s in S
        Term $w \leftarrow TermExtraction(s)$
        Assigning a entity type to term w using $NamedEntity(w)$
        if entity type is $PERSON$ AND $w \notin N$
            $cooc \leftarrow$ Measuring cooccurrence using $Overlap_{web}(e, w)$
            if $cooc > threshold$
                Adding w into a node List V
                Adding a edge between e and w into a edge List E
for each node v in V
    $ExtractSocialNetwork(n)$
return $(SN(e, V, E))$

Figure 6.1: Algorithm for extracting social networks

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.

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

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

of relations is greater than a certain threshold. Several indices can measure the co-occurrence [100]:

$$Matching_{web} = Hit(e_1 \cap e_2),$$

$$Dice_{web} = \frac{2 * Hit(e_1 \cap e_2)}{Hit(e_1) + Hit(e_2)},$$

$$Overlap_{web} = \frac{Hit(e_1 \cap e_2)}{min(Hit(e_1), Hit(e_2))},$$

$$Jaccard_{web} = \frac{Hit(e_1 \cap e_2)}{Hit(e_1) + Hit(e_2) - Hit(e_1 \cap e_2)},$$

$$PMI_{web} = \frac{Hit(e_1 \cap e_2) * N_{web}}{Hit(e_1) * Hit(e_2)},$$

where $e1$ and $e2$ are entities. N_{web} is the number of documents indexed by the search engine. In our case, we set $N = 10^{10}$ according to the number of indexed pages reported by Google. $Hit(e)$ is the hit counts retrieved by a query e.

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, the overlap coefficient [101] is used 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. The overlap coefficient is verified to perform well by investigating the probability of co-authorship [102]. We also employ the overlap coefficient.

The overlap coefficient is defined as

$$Overlap_{web}(e_1, e_2) = \begin{cases} \dfrac{Hit(e_1 \cap e_2)}{min(Hit(e_1), Hit(e_2))} & \text{if } Hit(e_1) > k \text{ and } Hit(e_2) > k, \\ 0 & \text{otherwise} \end{cases}$$

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

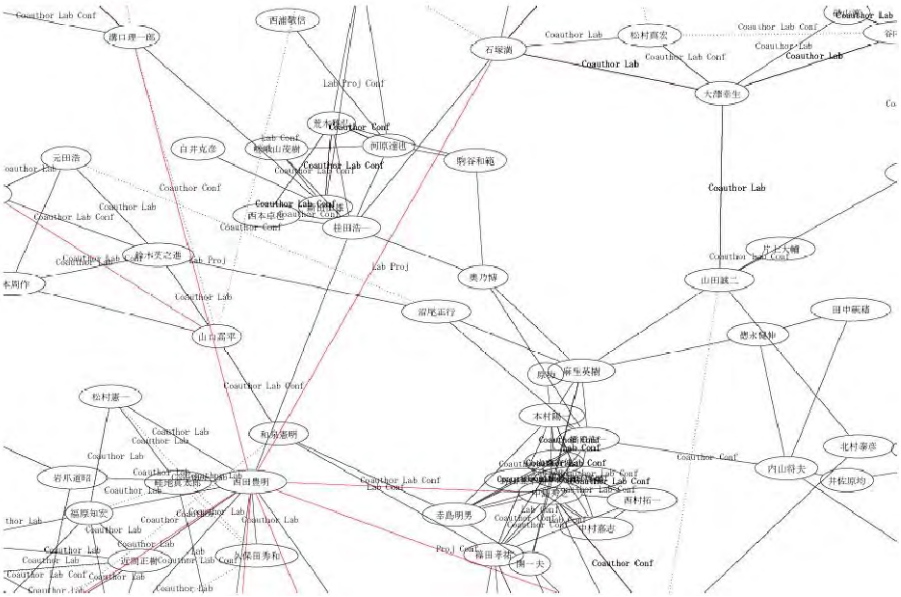

Figure 6.2: Part of the JSAI social network

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.

There is an alternative means to measure co-occurrence using a search engine, i.e., to use top retrieved documents. In this case, we can use the number of mentions and 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 by processing the documents. However, when the retrieved documents are much more numerous than k, we can process only a small fraction of the documents. Although various studies have applied co-occurrence by a search engine to extract a social network, most of them are summarised into the algorithm in Figure 6.1.

Figure 6.2 and 6.3 show the extracted social network of Japanese AI researcher using the proposed method.

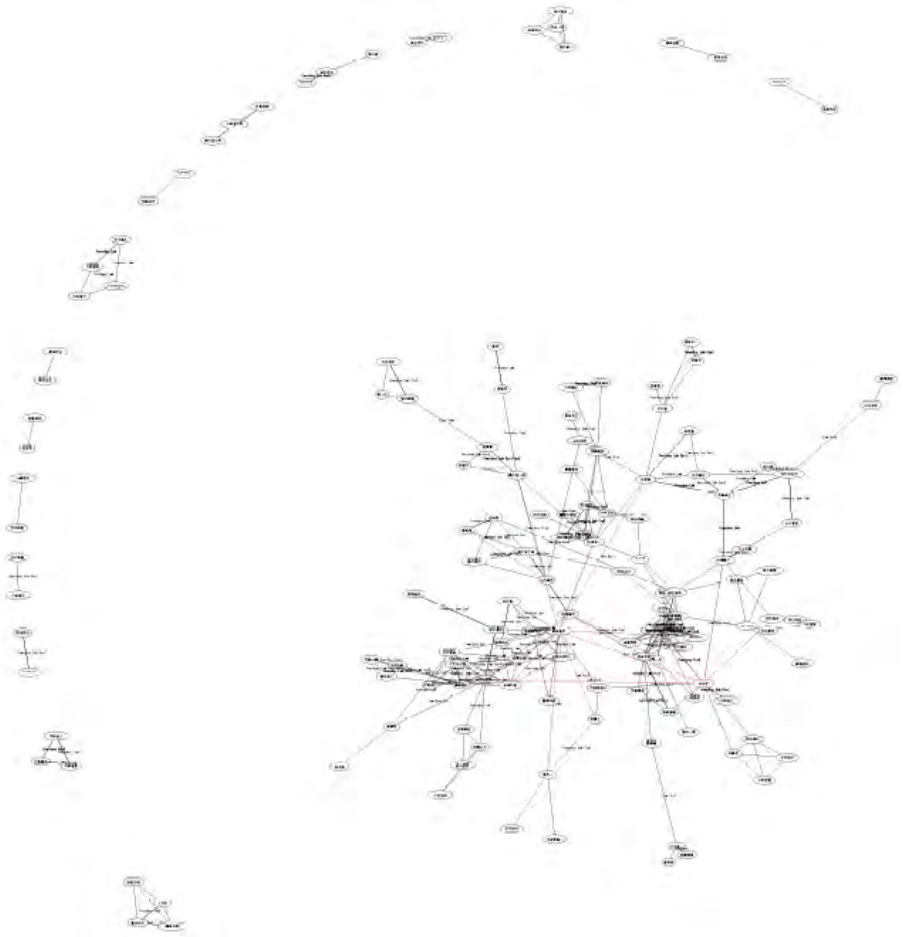

Figure 6.3: JSAI social network

Table 6.1: Keywords for "Mitsuru Ishizuka"

AI society, Yutaka Matsuo, Natural Language
Koichi Hashida, Hiroshi Dohi, Character Agent
Life-Like Interface, Naoaki Okazaki, University of Tokyo
Life-like Agent, AI journal

6.3.2 Node Information

Person names co-occur along with many words on the Web. For example, a particular researcher's name will co-occur with many words that are related to that person's major research topic. 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 [83].

The simple 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, we 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 that is introduced in Chapter 4 is used for the keyword extraction. Collecting documents retrieved by a person name, we obtain a set of words and phrases as candidates for keywords. We use Termex [108] 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 6.1 shows an example of keywords for Mitsuru Ishizuka. He have beenn working in the University of Tokyo and Japanese AI society; his research topics include Life-like agent and Natural Language Processing.

Table 6.2: Keywords for 6 kinds of relationships among Japanese AI researchers

1	co-authorship of conference paper	paper, author, conference venue, presentation, title, program
2	co-authorship of book	edit, book, publishing, programming, recommendation, co-author
3	co-edit of book	edit, revision, article, publishing, educational material, editor
4	collaborative project	representative person, contributor, minister, acceptance
5	co-authorship of journal paper	journal, Shogi, distribution, computer, information processing society of Japan
6	same affiliation	University of Tokyo, metropolitan, techonolgy, University, science

6.3.3 Edge Information

If we think about relations in the real-world, we find that various interpersonal relations exist: friends, colleagues, families, teammates, and so on. RELATIONSHIP [84] 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 [117]. It also provides interesting research topics recently in the context of complex networks [111].

Targetting the relations in a researcher community are targeted, Matsuo defined four kinds of relations according to the ease at identifying them and their importance in reality [138, 168].

- *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.

Matsuo propose a method to classifying th relation into these four kinds of relations using the supervised learning. They first fetch the top five pages retrieved by the *X AND Y* query. Then they extract features from the content of each page. Classifier such as Naive Bayes, maximum entropy or support vector machine is trained with the data of features and the classification rules are obtained. 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.

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 [109]. Relationships depend on the target domain; therefore, we must define classes to be categorized depending on a domain.

In contrast to the supervised learning, the unsupervised learning approach that requires neither a priori definition of relations nor preparation of large annotated corpora is possible. The method that is introduced in Chapter 5 is used as the unsupervised learning for extracting the relations in social networks. Table 6.2 showed the automatically extracted keywords for relations among Japanese AI researchers. The left column shows the relation labels that Mastuo defined. We found that extracted clusters and relation labels are corresponding those relations.

6.4 Discussion

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 [74, 88, 97, 98]. In addition, the natural language community has specifically addressed name disambiguation as a class of word sense disambiguation [116, 99].

Bekkerman and McCallum uses probabilistic models for the Web appearance disambiguation problem [74]: 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 [96]. 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.

Matsuo and Bollegara developed a name-disambiguation method [75]. 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.

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.

In fact 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. How can we reduce the number of queries while maintaining the extraction performance? One idea is to filter out pairs of persons that seem to have no relation. 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]

6.4.1 Future Trends

Social network extraction has some concrete applications for Semantic Web and information retrieval. It also contributes as a more general data mining potentially.

One of the possible directions is to expand the applicability to other named entities such as cooperates, organizations, books, and so on. Actually, some studies try such expantion [91]. Because there are lots of entities which are interests of Web users, mining the structure and show the overview is a promising application: It helps users decision making and information gathering. Depending on the target entities, appropriate methods will vary: In case of researchers, the Web count is useful to detect the strength of ties among them, while in corporates case, the Web count does not produce a good index. Some relations of corporates are paid much more attentions than others, thus returns huge hit counts. Therefore more advanced language processing is necessary to identify the individual different relations.

Increasing number of studies are done that uses a search engine. In natural language processing research, much works begin using a search engine. For example, F. Keller et al. [178] use the web to obtain frequencies for unseen bigrams in a given corpus. They count for adjective-noun, noun-noun, and verb-object bigrams by querying a search engine, and demonstrate that web frequencies (web counts) correlate with frequencies from a carefully edited corpus such as British National Corpus (BNC). Besides counting bigrams, various tasks are attainable by web-based models: spelling

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

Table 6.3: Number of participants at conferences.

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

correction, adjective ordering, compound noun bracketing, countability detection and so on [178]. For some tasks, simple unsupervised models outperform better when n-gram frequencies are obtained from the web rather than a standard large corpus; the web yields better counts than BNC. In the future, there will be more and more studies using a search engine, which can be considered as a database interface to the huge amount of information on social and linguistic activities.

6.5 Application: Researcher Search System using Social Networks

To demonstrate our Web mining approach in the real application, we develop a researcher mining and retrieval system called Polyphonet. The system is an example of an end-user application that integrates Web mining into the Semantic Web. The system has been operated at the 17th–21st Annual Conferences of the Japan Society of Artificial Intelligence (JSAI2003–2007) 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. POLYPHONET is incorporated with a scheduling support system [89] and a location information display system [110] in the ubiquitous computing environment at the conference sites. 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.

The system is intended to provide a search function based on the relation of

researchers and promote efficient collaboration. A social network of participants is displayed in the 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. For example, a user can find what research topic a researcher is doing or whom she is working with. Social networks is used for finding path to other researchers or recommending related researchers. If the researcher is not found in the system, a user can register his name. Subsequently, the system automatically extracts information from the Web using the proposed Web mining method.

Figure 6.4 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 6.5 shows the obtained shortest path between two persons on a social network. Figure 6.6 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 6.3. Comments were almost entirely positive; they enjoyed using the system.

Extracted users' information is easily incorporated in the RDF representation [127]. For example, the network ties and the interest associations are represented in RDF using the `foaf:knows` and `foaf:interst` properties. Similarly, the relation become `foaf:Persons` with the appropriate relations. Some extensions of the FOAF model are necessary for expressing the relation labels. Figure 4 shows a FOAF file that was generated based on extracted information. Each researcher can have metadata included in the system. because extracted information is stored as a FOAF file.

Trust gives an authoritativeness of a person which is useful when finding an important researcher in the field. If we trace the node which has high individual trust from antecedent node, we can find the circle of trust which comprises the small "Web of Trust" in a community.

6.6 Conclusions

This chapeter describes a method for extracting social network from the Web. The Web is currently a huge source of information for the relation between entities. Our method leverages co-occurrence information obtained from a search engine to extract a social network among entities. As an actual application that utilizes the proposed method, we presented a researcher search system.

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 becoming a popular concept. We intend to apply our approach in the future to extract much structural knowledge aside from social networks.

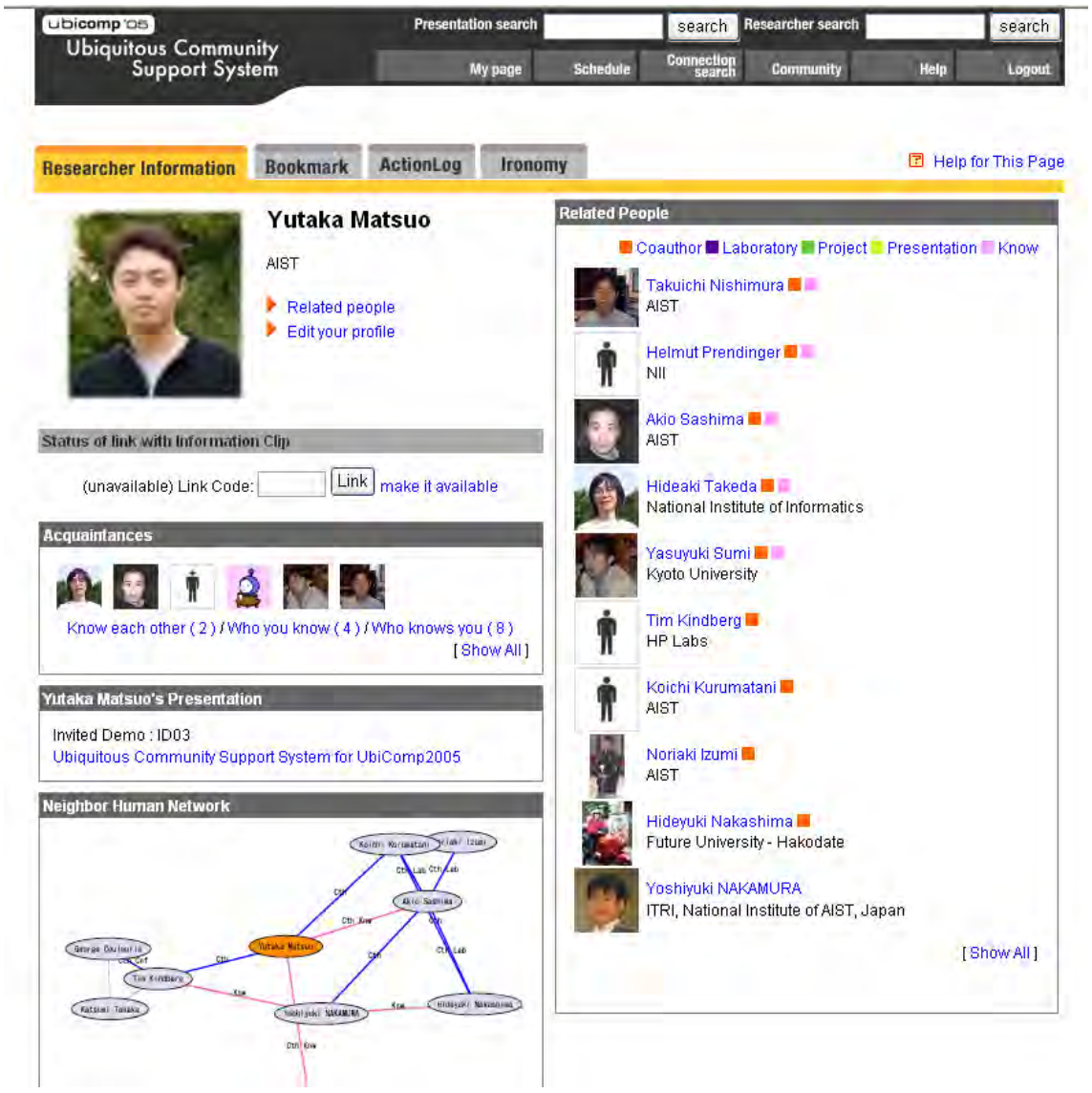

Figure 6.4: My page on POLYPHONET

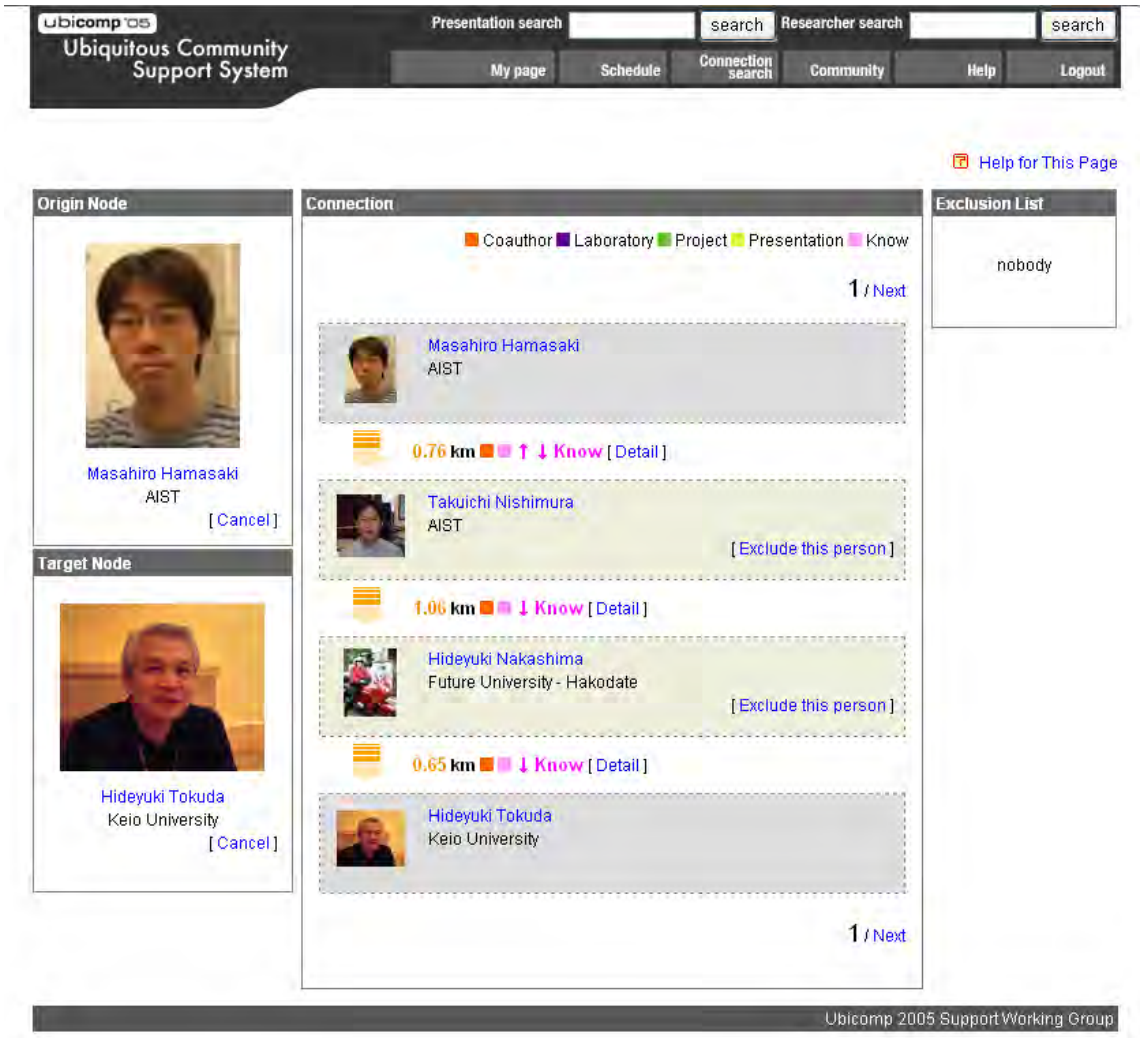

Figure 6.5: Shortest path from a person to a person on POLYPHONET

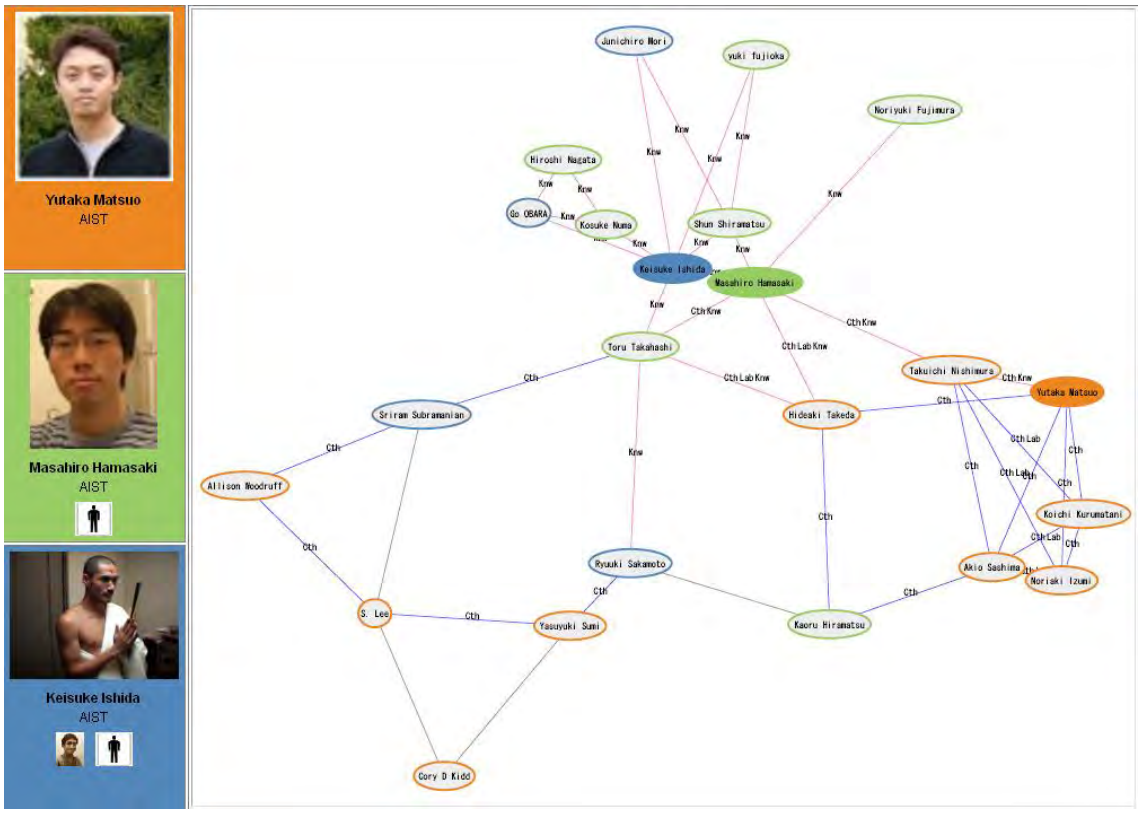

Figure 6.6: Social network among three persons on POLYPHONET

```
<rdf:RDF
xmlns:rdf=''http://www.w3.org/1999/02/22-rdf-syntax-ns#''
xmlns:foaf=''http://xmlns.com/foaf/0.1''
xmlns:acsn=''http://www.carc.aist.go.jp/ y.matsuo/acsn/0.1''>
<foaf:Person>
<foaf:mbox rdf:resource="ishizuka@miv.t.u-tokyo.ac.jp"/>
<foaf:name>Mitsuru Ishizuka</foaf:name>
<foaf:interest rdfs:label="Character agent"
rdf:resource="http://www.miv.t.u-tokyo.ac.jp"/>
<foaf:currentProject rdfs:label ="Life-like interface"
rdf:resource="http://www.miv.t.u-tokyo.ac.jp"/>
<foaf:workplaceHomepage rdfs:label="University of Tokyo"
rdf:resource="http://www.miv.t.u-tokyo.ac.jp"/>
<acsn:Coauthor>
<foaf:Person>
<foaf:mbox rdf:resource="y.matsuo@aist.go.jp"/>
<foaf:name>Yutaka Matsuo</foaf:name>
</foaf:Person>
</acsn:Coauthor>
</foaf:Person>
```

Figure 6.7: An example of a FOAF file tha tis based on extracted information from the Web.

Chapter 7

Information Sharing using Social Networks

While users disseminate various information in the open and widely distributed environment of the Semantic Web, determination of who shares access to particular information is at the center of looming privacy concerns. We propose a real-world-oriented information sharing system that uses social networks. The system automatically obtains users' social relationships by mining various external sources. It also enables users to analyze their social networks to provide awareness of the information dissemination process. Users can determine who has access to particular information based on the social relationships and network analysis.

7.1 Introduction

With the current development of tools and sites that enable users to create Web content, users have become able to easily disseminate various information. For example, users create Weblogs, which are diary-like sites that include various public and private information. Furthermore, the past year has witnessed the emergence of social networking sites that allow users to maintain an online network of friends or associates for social or business purposes. Therein, data related to millions of people and their relationships are publicly available on the Web.

Although these tools and sites enable users to easily disseminate information on the Web, users sometimes have difficulty in sharing information with the right people and frequently have privacy concerns because it is difficult to determine who has access to particular information on such applications. Some tools and applications provide control over information access. For example, Friendster, a huge social networking site, offers several levels of control from "public information" to "only for friends". However, it provides only limited support for access control.

An appropriate information sharing system that enables all users to control the dissemination of their information is needed to use tools and sites such as Weblog, Wiki, and social networking services fully as an infrastructure of disseminating and sharing information. In the absence of such a system, a user would feel unsafe and would therefore be discouraged from disseminating information.

How can we realize such an information sharing system on the Web? One clue

exists in the information sharing processes of the real world. Information availability is often closely guarded and shared only with the people of one's social relationships. Confidential project documents which have limited distribution within a division of company, might be made accessible to other colleagues who are concerned with the project. Private family photographs might be shared not only with relatives, but also with close friends. A professor might access a private research report of her student. We find that social relationships play an important role in the process of disseminating and receiving information. This chapter presents a real-world oriented information sharing system using social networks. It enables users to control the information dissemination process within social networks.

The remainder of this chapter is organized as follows: section 2 describes the proposed information sharing system using social networks. In section 3, we describe the application of our system. Finally, we conclude this chapter in section 4.

7.2 Information Sharing using Social Networks

Figure 7.1 depicts the architecture of the proposed information sharing system. The system functions as a "plug-in" for applications so that external applications enable users to leverage social networks to manage their information dissemination. A user can attach an access control list to his content using his social network when creating content on an application. Then, when the application receives a request to access the content, it determines whether to grant the request based on the access control list.

Because users determine the access control to information based on the social network, the system requires social network data. The system obtains users' social networks automatically by mining various external sources such as Web, emails, and sensor information; subsequently, it maintains a database of the social network information. Users can adjust the network if necessary.

The system enables users to analyze their social network to provide awareness

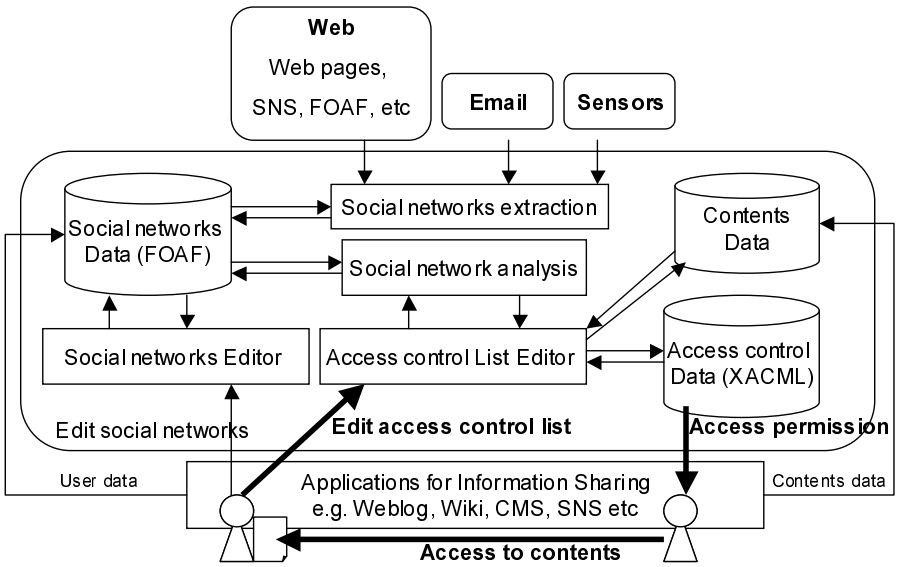

Figure 7.1: Architecture of the proposed information sharing system

of the information dissemination process within the social network. Using social relationships and the results of social network analyses, users can decide who can access their information.

Currently, the proposed system is applied to an academic society because researchers have various social relationships (e.g., from a student to a professor, from a company to a university) through their activities such as meetings, projects, and conferences. Importantly, they often need to share various information such as papers, ideas, reports, and schedules. Sometimes, such information includes private or confidential information that ought only to be shared with appropriate people. In addition, researchers have an interest in managing the information availability of their social relationships. The information of social relationships of an academic society, in particular computer science, is easily available online to a great degree. Such information is important to obtain social networks automatically.

Hereafter, we explain in detail how social networks are modeled, extracted and analyzed. Then we explain how users can decide to control information access using social networks.

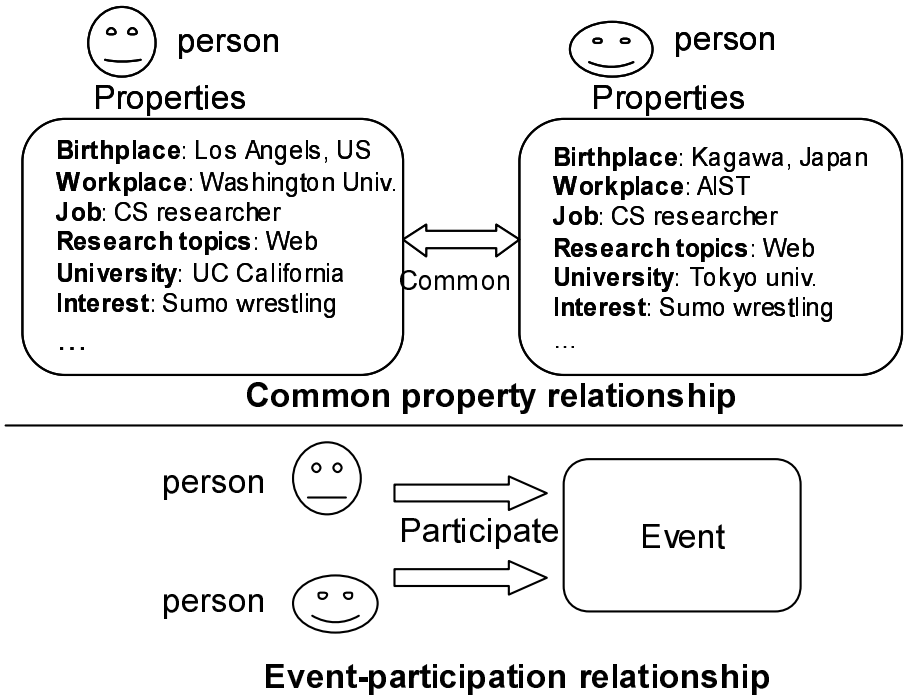

Figure 7.2: Two kinds of relationships

7.2.1 Representation of Social Relationships

With the variety of social relationships that exist in the real world, a salient problem has surfaced: integration and consolidation on a semantic basis. The representation of social relationships must be sufficiently fine-grained that we can capture all details from individual sources of information in a way that these can be recombined later and taken as evidence of a certain relationship.

Several representations of social relationships exist. For example, social network sites often simplify the relationship as "friend" or "acquaintance". In the Friend of a Friend (FOAF) [169] vocabulary, which is one of the Semantic Web's largest and most popular ontologies for describing people and whom they know, many kinds of relationships between people are deliberately simplified as "knows" relations. A rich ontological consideration of social relationships is needed for characterization and analysis of individual social networks.

We define two kinds of social relationship (Fig. 7.2) [175]. The first basic structure of social relationship is a person's participation in an event. Social relationships come into existence through events involving two or more individuals. Such events might not require personal contact, but they must involve social interaction. From this event, social relationships begin a lifecycle of their own, during which the characteristics of the relationship might change through interaction or the lack thereof. An event is classified as *perdurant* in the DOLCE ontology [174], which is a popular ontology. For example, an event might be a meeting, a conference, a baseball game, a walk, etc. Assume that person A and person B participate in Event X. In that situation, we note that A and B share an *event co-participation relationship* under event X.

A social relationship might have various social roles associated with it. For example, a student-professor relationship within a university setting includes an individual playing the role of a professor; another individual plays the role of a student. If A and B take the same role to Event X, they are in a *same role relationship* under event X (e.g., students at a class, colleagues in a workspace). If A cannot take over B's role or vice versa, A and B are in a *role-sharing relationship* (e.g., a professor and students, a project leader and staff).

Another kind of social relationship is called a *common property relationship*. Sharing the same property value generates a common property relationship between people. For example, person A and person B have a common working place, common interests, and common experiences. Consequently, they are in a common property relationship with regard to those common properties.

7.2.2 Extraction of Social Networks

If two persons are in either an event co-participation relationship or a common property relationship, they often communicate. The communication media can be diverse: face-to-face conversation, telephone call, email, chat, online communication on Weblogs, and so on. If we wish to discover the social relationship by observation, we must estimate relationships from superficial communication. The emerging field of

social network mining provides methods for discovering social interactions and networks from legacy sources such as web pages, databases, mailing lists, and personal emails.

Currently, we use three kinds of information sources to obtain social relationships using mining techniques. From the Web, we extract social networks using a search engine and the co-occurrence of two persons' names on the Web. Consequently, we can determine the following relationships among researchers: Coauthor, Same affiliation, Same project, Same event (participants of the same conference, workshop, etc.) [176]. Coauthor and Same event correspond to an event co-participation relationship. Same affiliation and same project correspond to a common property relationship. We are also using other sources such as email and sensors (we are developing a device that detects users within social spaces such as parties and conferences) to obtain social relationships.

Necessarily, the quality of information obtained by mining is expected to be inferior to that of manually authored profiles. We can reuse those data if a user has already declared his relationships in FOAF or profiles of social networking services. Although users might find it difficult and demanding to record social relations, it would be beneficial to ask users to provide information to obtain social relationships.

In addition to the relationship type, another factor of the social relationship is tie strength. Tie strength itself is a complex construct of several characteristics of social relations. It is definable as affective, frequency, trust, complementarity, etc. No consensus for defining and measuring them exists, which means that people use different elicitation methods when it comes to determining tie strength. For example, Orkut, a huge social networking service, allows description of the strength of friendship relations on a five-point scale from "haven't met" to "best friend", whereas other sites might choose other scales or terms.

In our system, we use trust as a parameter of tie strength. Trust has several very specific definitions. In [172], Golbeck describes trust as credibility or reliability in a human sense: "how much credence should I give to what this person speaks about" and "based on what my friends say, how much should I trust this new person?" In the context of information sharing, trust can be regarded as reliability regarding "how

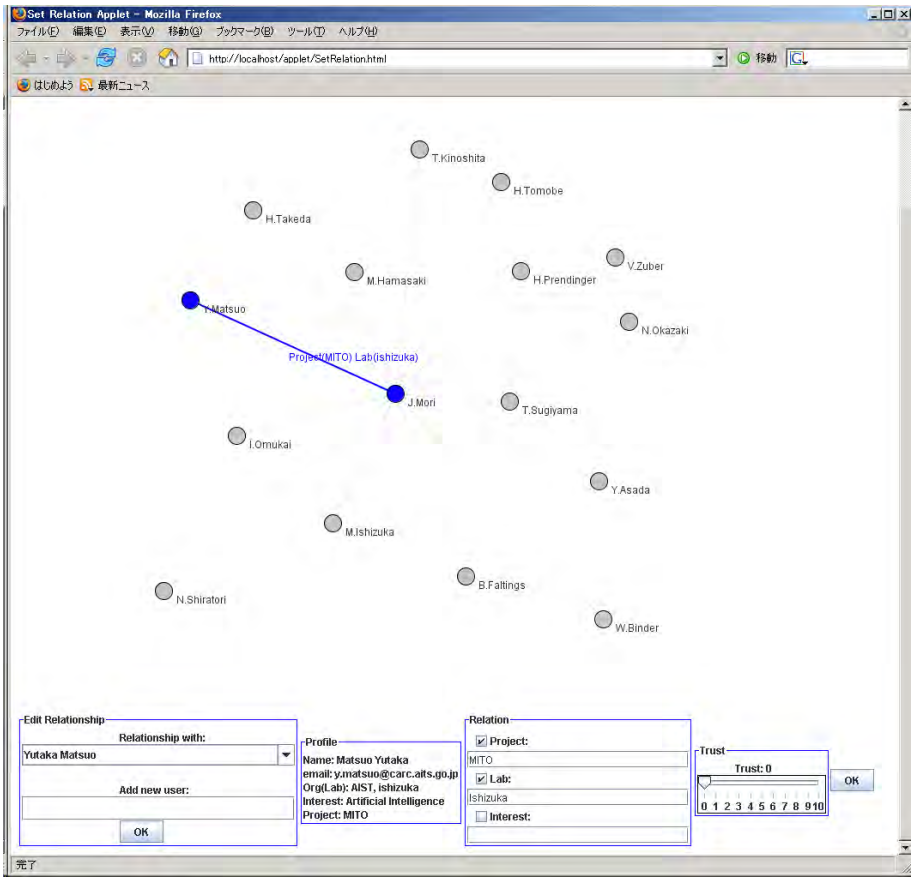

Figure 7.3: Editor for social relationships

a person will handle my information". Users can give trust directly in a numerical value to a person in his relation. Alternatively, trust is obtainable automatically as authoritativeness of each person using the social network [176].

The obtained social network data are integrated as extended FOAF files and stored in database. Users can adjust networks if needed (Fig. 7.3). The social relationship and its tie strength become guiding principles when a user determines an access control list to information.

7.2.3 Social Network Analysis for Information Sharing

The system enables users to analyze their social networks to provide awareness of the information dissemination process within the social network.

Social network analysis (SNA) is distinguishable from other fields of sociology by its focus on relationships between actors rather than attributes of actors, a network view, and a belief that structure affects substantive outcomes. Because an actor's position in a network affects information dissemination, SNA provides an important implication for information sharing on the social network. For example, occupying a favored position means that the actor will have better access to information, resources, and social support.

The SNA models are based on graphs, with graph measures, such as centrality, that are defined using a sociological interpretation of graph structure. Freeman proposes numerous ways to measure centrality [170]. Considering a social network of actors, the simplest measure is to count the number of others with whom an actor maintains relations. The actor with the most connections, the highest degree, is most central. This measure is called *degreeness*. Another measure is *closeness*, which calculates the distance from each actor in the network to every other actor based on connections among all network members. Central actors are closer to all others than are other actors. A third measure is *betweenness*, which examines the extent to which an actor is situated among others in the network, the extent to which information must pass through them to get to others, and consequently, the extent to which they are exposed to information circulation within the network. If the betweenness of an actor is high, it frequently acts as a local bridge that connects the individual to other actors outside a group. In terms of network ties, this kind of bridge is well known as Granovetter's "weak tie" [173], which contrasts with "strong tie" within a densely-closed group.

As the weak tie becomes a bridge between different groups, a large community often breaks up to a set of closely knit group of individuals, woven together more loosely according to occasional interaction among groups. Based on this theory, social network analysis offers a number of clustering algorithms for identifying communities based on network data.

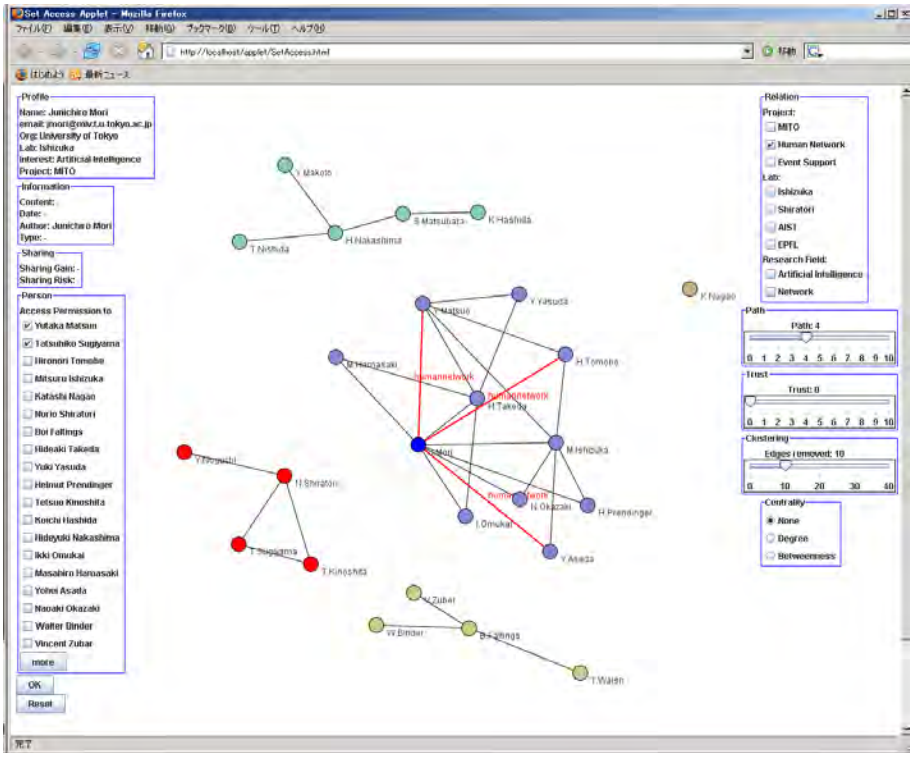

Figure 7.4: Editor for analyzing social networks and assigning an access control list to content

The system provides users with these network analyses (Fig. 7.4) so that they can decide who can access their information. For example, if user wants to diffuse her information, she might consider granting access to a person (with certain trust) who has both high degreeness and betweenness. On the other hand, she must be aware of betweenness when the information is private or confidential. Clustering is useful when a user wishes to share information within a certain group.

7.3 Application

To demonstrate and evaluate our system, we developed a community site (Fig. 7.5) using communication tools such as Weblogs, Wikis, and Forums. By that system,

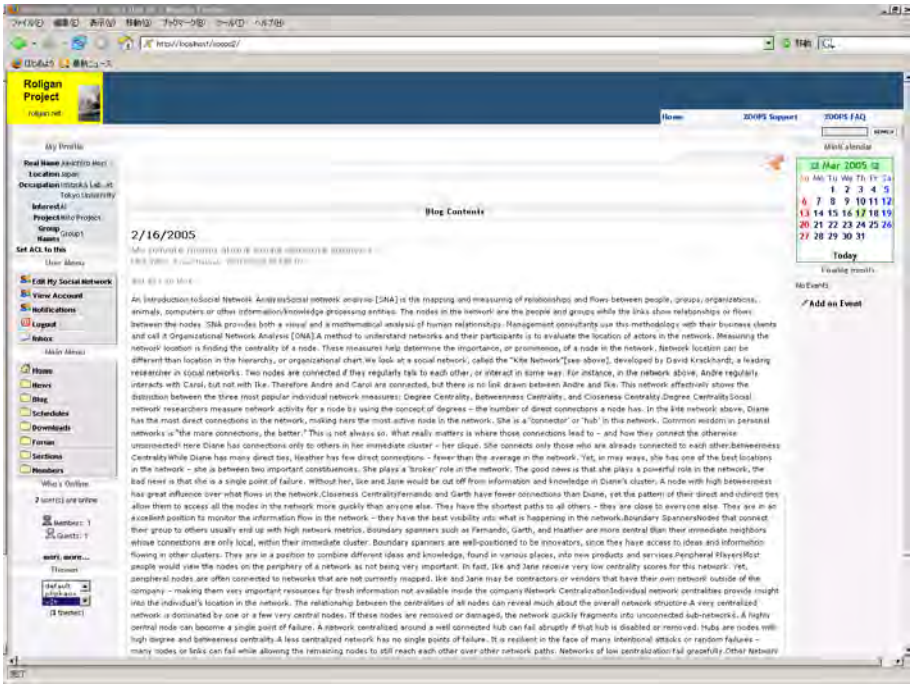

Figure 7.5: Web site for sharing research information

studies from different organizations and projects can be disseminated and their information thereby shared. Users can share various information such as papers, ideas, reports, and schedules at the site. Our system is integrated into a site that provides access control to that information. Integrating our system takes advantage of the open and information nature of the communication tools. It also maintains the privacy of the content and activities of those applications.

Users can manage their social networks (Fig. 7.3) and attach the access control list to their content (e.g., Blog entries, profiles, and Wiki pages) using extracted social relationships and social network analysis (Fig. 7.4).

Once a user determines the access control list, she can save it as her information access policy for corresponding content. The access policy is described using extended eXtensible Access Control Markup Language (XACML) and is stored in a database. She can reuse and modify the previous policy if she subsequently creates a similar content.

One feature of our system is that it is easily adaptable to new applications because of its plug-and-play design. We are planning to integrate it into various Web sites and applications such as social network sites and RSS readers.

7.4 Related Works and Conclusions

Goecks and Mynatt propose a Saori infrastructure that also uses social networks for information sharing [171]. They obtain social networks from users' email messages and provide sharing policies based on the type of information. We obtain social networks from various sources and integrate them into FOAF files. This facilitates the importation and maintenance of social network data. Another feature is that our system enables users to analyze their social networks. Thereby, users can control information dissemination more effectively and flexibly than through the use of predefined policies.

As users increasingly disseminate their information on the Web, privacy concerns demand that access to particular information be limited. We propose a real-world oriented information sharing system using social networks. It enables users to control the information dissemination process within social networks, just as they are in the real world. Future studies will evaluate the system with regard to how it contributes to wider and safer information sharing than it would otherwise. We will also develop a distributed system that can be used fully on the current Web.

Chapter 8

Expert Finding using Social Networks

Recent advances of the Web and ubiquitous environment enable users to accumulate and share their experiences. Since humans usually also take others' experiences into account for decision making, an intriguing extension of this idea is to assist users in the sharing of such experiences. One important issue in sharing experience is to select relevant sharing partners who have appropriate knowledge and information on current specific topics. We propose a method which employs the user profile and social structure of a Web community in order to find sharing partners who have appropriate expertise and are likely to be able to reply to a request. We addressed the issue in the scenario from the actual social network service for sharing recipes. Utilizing the user information and social structure from the existing Web community, we implemented and operate the community mining system which locates relevant and socially close experts for information seekers.

8.1 Introduction

Human decision making usually takes not only the decision maker's personal experiences into account, but also experiences and opinions of other persons. This behavior can be supported by a personal assistant, which has access to experiences of people known to the user. Actually, the recent Web communities such as blogs and social network services provides huge collections of experiences which could be exploited for realizing such support. In addition to the Web communities, recent development of ubiquitous technologies also enable users to share experiences by automatically capturing and recording their actions. The data from ubiquitous environments may be recorded in a way easy to process by machines, therefore it can also be exploited for ubiquitous user support (e.g., recommenders, teaching systems) beneficial for recipients of such experiences.

Although large amount of user experiences from the Web and the ubiquitous environments are currently available, there is still the question of *whose* experiences should be applied for support. Candidates can be found in various places, for instance, in the user's previous contacts, among people physically nearby, or on the Web. Their count might be large, which suggests to support the user explicitly in this selection

problem. Here, we propose to exploit that people interested in some domain are frequently organized in a Web-based community—which might hold for the candidates as well. Thus, an analysis of a candidate's social relationships within a community matching the user's problem is a promising way of not only estimating expertise—but also of the candidate's will to assist the user.

In this chapter, aiming at helping a user find relevant sharing partners of experiences, we propose the method for mining Web communities. In particular we address the issue of finding experts from the Web community who have appropriate knowledge and information on specific topics. We focus on the scenario from the actual social network service for sharing recipes. Utilizing the user information and social structure from the existing Web community, we implemented and operate the community mining system which locates relevant and socially close experts for information seekers.

In the next Section 2, we compare our approach with related research. We continue in Section 3 with a description of our example scenario. Then, in Section 4 we explain in details about the proposed method to find experts in the Web community. Finally, we conclude the chapter with an outlook on future work in Section 5.

8.2 Related Work

One of the central questions addressed in this chapter is how to find relevant sharing partners of experiences, who might be able to answer a given information need. This issue is closely related to the recommendation community members in social matching systems (cf. [67]). For instance, an expertise recommender [47] may help members of an organization to locate other members who have specific expertise. Such expertise finders have been explored in a series of studies [66, 2]. Here, systems such as Referral Web [38], ER [47], and MARS [70] leveraged social networks as a means of finding people.

Most of these expert finding systems mainly focus on specific domains within organizations. However, there is a growing interest in exploiting ubiquitous information for this task: recent research has shown that social networks and communities could

also be obtained from the sensor information [34]. In addition, due to the popularity of blogging and social network services, a tremendous amount of sharing-related information is becoming available online. Thus, the task of expert finding now becomes the problem not only for specific users but also for every single end user. Thereby, the research question which should be newly addressed in this area is how to find appropriate sharing partners for end users by using the ubiquitous information and other information from the Web.

Social visualization systems (e.g., [65, 27]) offer rich graphical representations of a community's social activity to support a user in finding someone to communicate with. However, most social visualizations only highlight personal contacts or represent the exchange of information. They rarely address the issue of finding appropriate users for a given information need. Reputation systems such as ebay and expertise finders highlight users who may be reliable or experts in general in a domain, which is of interest for our system, but recommendations are usually the result of algorithms which are due to their complexity hidden from the user. Consequently, those mechanisms cannot be controlled and influenced by the user. Therefore, with respect to the variety of situations in which the user may need community members' experiences in our case, we researched on flexible ways to allow the user to specify which kinds of persons are likely to be of interest in these particular situations.

8.3 Example Scenario: Finding Experts in Recipe Sharing

There currently exist many online communities aiming at information sharing on the Web, where users can share their interests, maintain their relationships and communicate with each other. Among recent online communities, social networking services (SNSs) have received much attention on the Web. SNSs enable users to register their friends. Therein, the users can create their contents such as profiles and Blogs and communicate with their friends. One important feature on SNSs is information sharing because information on SNSs such as private profiles, photos, and Blogs are

Figure 8.1: Bakespace.

neither completely open nor closed: they can be shared loosely among user's friends, colleagues and acquaintances with elaborate access control. For sharing various information, several SNSs have increasingly emerged targeting niche communities such as music, medical, cooking, and business communities.

Recipe information is one of actively shared experiences in these SNSs. Users are eager to share their recipes and try to find relevant sharing partners. Recipe often involves complex constraints such as availability of ingredients, food allergies, dietary rules, and religious food preferences. Thus, finding relevant experts are important for the users. However, it is difficult for the users to find experts from the SNSs just by looking at each Web page. For example, suppose one user is looking for someone to

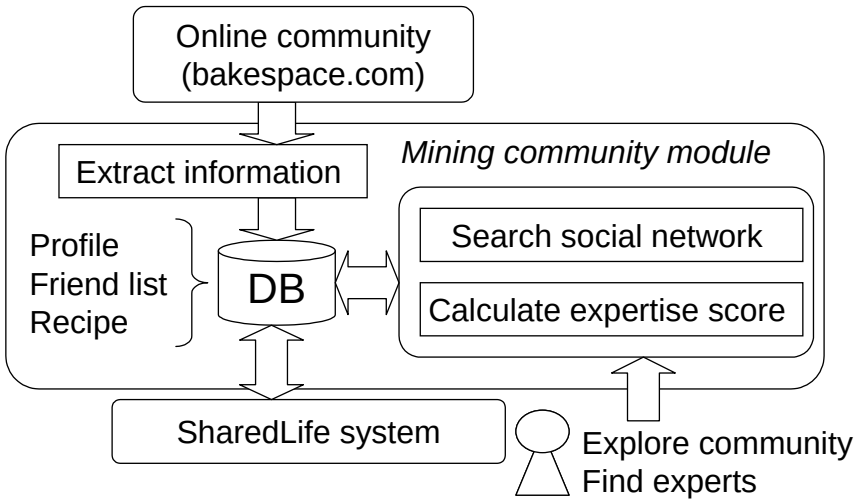

Figure 8.2: System Overview of Mining Community Module.

ask advices about vegetarian foods and recipes. The user has to manually check who is creating vegetarian recipes from the recipe list. Even if the user could find a right expert to ask advices, the user is not sure whether the expert is socially close and therefore is likely to answer the question.

In order to address this issue of finding experts in the Web community for sharing recipes, we focus on BakeSpace (Fig. 1) [1] which is a Web community that combines recipe sharing with comprehensive social networking. Users on BakeSpace can create their profiles and blogs, make new friend and share recipes with other users. In the following, we will explain in details about how to mine the Bakespace community and find appropriate experts who could be potential sharing partners.

8.4 Expert Finding in Social Networks

8.4.1 System Overview

Figure 8.2 show an overview of the proposed system for mining community (mining community module). This mining community module works as an external add-on

[1] http://bakespace.com

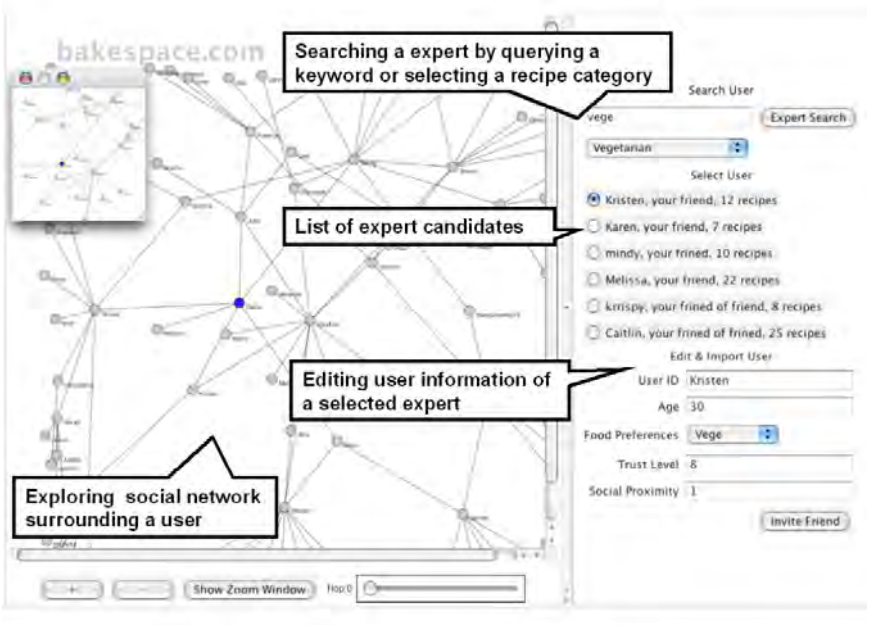

Figure 8.3: User Interface of Mining Community Module.

module of existing Web communities. The system first extracts information from external online communities by processing html pages. In our example, the system extract user profiles, friend lists, and recipe information from BakeSpace. Extracted information are stored in the database of mining community module. When the system receives a query from a user who tries to find experts, the system searches the user's social network on BakeSpace and compute the expertise of neighboring users. The expertise is scored according to relevancy to the given query. Finally the system returns the list of expert candidates and shows the referral chain which is a network path from the user to the expert.

Now we go in details about our method to find experts. In our example scenario, a user is not familiar with vegetarian foods and recipes therefore she would like to find an expert who could share some useful experience about vegetarian stuff and answer some questions. In addition she would like to find someone who is socially close to her (e.g., a friend of a friend) so that she can easily contact a expert. To achieve these goals, we now address following two questions: (1) Who are the experts

on a certain topic (Expert Model) and (2) How the experts can be accessed (Search Social Networks).

8.4.2 Expert Model

Our expert model is based on probabilistic language model [7] which has been successfully applied in many Information Retrieval tasks. In our expert finding task, when the system is given a topic q (e.g. a recipe category such as meat and vegetable), it returns a list of candidate experts which are ranked according to their expertise on q. For this, we calculate the probability $p(u|q)$ that a candidate u is an expert given the query topic q. And we rank the candidates according to this probability. The top candidates are deemed the most probable experts for the given query. Using the Bayes' Theorem, we compute the probability $p(u|q)$ as

$$p(u|q) = \frac{p(q|u)p(u)}{p(q)},$$

where $p(u)$ is the probability of a candidate and $p(q)$ is the probability of a query and $p(q|u)$ the probability of the query given the candidate. $p(u)$ is estimated as the number of recipes that a candidate u has created in Bakespace. $p(q)$ is estimated as the number of recipes that are categorized into a recipe category q (Bakespace provides recipe categories and every recipe is categorized into one of the categories). $p(q|u)$ is estimated as the number of recipes that are created by a candidate u and categorized into a recipe category q.

8.4.3 Search Social Networks

To find a expert who is socially close to a user, the system searches his or her social network. Following the classic study by Travers and Milgram about the "small world" and "six degree of separation" [51], several studies have addressed the methods for searching social networks, which can also be adapted to locating experts. Adamic proposes best connected search (BCS) algorithm which makes use of the skewed degree distribution of many networks [4]. Breadth First Search (BFS) searches a user's ego

centric network by starting from ego and expanding its search to neighboring nodes along with the network paths. Because BFS has the strength of finding the target closest to the source, which matches our requirement to find socially close experts, we employ BFS for searching social networks.

Searching social networks according to BFS, we also consider the network centrality of a expert. In social network analysis (SNA) several ways to measure centrality for socialogical interpretation of network structure have been proposed [26]. The simplest measure, called degreeness, is to count the number of links that each node has. We employ this degreeness to estimate the accessibility to a expert. Here, our assumption is that if a expert has less links, he or she can be more accessible than other experts who have many links therefore are overloaded with many seekers. In summary our algorithm of searching social network find a expert e in the list of expert candidates such that

$$\arg\min_{e} \; path(e)\; degreeness(e),$$

where $path(e)$ is the length of network path from a user to a expert e and $degreeness(e)$ is the degree centrality of a expert e.

Figure 8.3 shows the user interface of our mining community module. A user can find experts by querying keywords or choosing recipe categories. The system returns the list of expert candidates and some evidence such as the link to expert's page in Bakespace and the number of recipes. After selecting the expert, the user can see additional information about the expert and contact the expert.

Using the interface, a user can explore his or her social network. Visualising the social network help the user contact a expert by providing the referral chain which is a network path from the user to the expert. This can also be used by the user both to assess the credibility of the expert and as a source of people who might introduce the seeker to the expert. As for supporting the user to communicate with the experts, the user can use the message system in the site or other external interaction means such as emails and Instant Messengers.

8.5 Conclusion and Future Work

Recent advances of the Web and ubiquitous environment enable users to accumulate and share their experiences. By sharing such augmented personal memories, users can be supported in exchanging opinions, guiding others, or just telling stories One important issue in sharing experience is to select relevant sharing partners who have appropriate knowledge and information on current specific topics. In this chapter, we reported about our ongoing research efforts towards expert finding in social networks. Our method employs the user profile and social structure of a Web community in order to find sharing partners who have appropriate expertise and are likely to be able to reply to a request. We addressed the issue in the scenario from the actual social network service for sharing recipes. Utilizing the user information and social structure from the existing Web community, we showed our implementation of community mining system which locates relevant and socially close experts for information seekers.

In the future, we will extend the presented work in ubiquitous environments. Social relations among users are currently available not only from the Web information but also ubiquitous information such as location data. Our approach for finding experts in social networks could be applied to help a user find relevant sharing partners of experiences in ubiquitous environments

Chapter 9

Conclusion

With the large quantity of information that users publish on the Web, the Web has now turned to the huge corpus that can be easily accessible using search engines, which opens new possibility to handle the vast relevant information and mine important structures and knowledge. As users publish their daily activities and communicate in the Web, the Web is now becoming another form of our society. Therein, eliciting and representing entity information is increasingly important to link the Web to the real world. In particular, with the currently growing trend toward the Semantic Web, extracting information about entities and the relations among them has been gained interest.

This thesis proposed novel methods for extracting entity information and entity relations from the Web. The key features of our approach are to leverage existent search engine and obtain several Web-scale statics such as hit counts and snippets in order to assess entity-related information. Applying several text processing techniques such as named entity recognition and clustering to the information obtained from search engine, our methods extract entity information, entity relations and social networks. The extracted information can be applied to several applications. We developed the researcher search system in which the information about researchers and relationships are automatically extracted from the Web. We also developed the information sharing system and the expert finding system using the extracted social networks.

Overall, in this thesis we address two research questions for extracting entity information from the Web: (1) how search engines can be used to access the Web corpus and extract entity information from the Web and (2) how the extracted entity information can be used to support users in entity-based information services.

For first question, we propose a method of keyword extraction for extracting entity information from the Web. The proposed method is based on the statistical features of word co-occurrence that are obtained from search engine. We also propose a method that extracts descriptive labels of relations among entities automatically such as affiliations, roles, locations, part-whole, social relationships. Fundamentally, the method clusters similar entity pairs according to their collective contexts obtained from search engine. The descriptive labels for relations are obtained from the results of

clustering. Finally, We propose a method that automatically extracts social networks from the Web. The method leverages search engine to build social networks by merging the information distributed on the Web.

For second question, we develop the researcher search system. The system is a Web-based system for an academic community to facilitate communication and mutual understanding based on entity information and social networks extracted from the Web. We also develop a real-world-oriented information sharing system. The system enables users to determine who has access to particular information based on the social networks and network analysis. We finally develop the expert finding system which locates relevant and socially close experts for information seekers. The system leverages the entity information and social networks of a Web community in order to find experts who have appropriate expertise.

Importantly, our approach use the Web as huge database and a search engine as its interface to obtain entity information. Recent important approaches of a Web mining toward the Semantic Web use the Web as a huge language corpus and combine with a search engine. The underlying concept of these methods is that it uses globally available Web data and structures to annotate local resources semantically to bootstrap the Semantic Web. Merging the vast amount of information on the Web and producing higher-level information might contribute many knowledge-based systems in the future.

Future studies will explore the possibilities of extending the proposed method to several types of entities and relations. Enriching entities information extracted from the Web automatically, our method might contribute to several information services such as information retrieval, question answering, summarisation, and recommendation.

Bibliography

[1] 2001. Representing vCard Objects in RDF/XML, http://www.w3.org/TR/2001/NOTE-vcard-rdf-20010222/.

[2] M. S. Ackerman, V. Pipek, and V. Wulf. *Sharing Expertise: Beyond Knowledge Management*. MIT Press, Cambridge MA, 2003.

[3] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 23(3), 2003.

[4] L. A. Adamic and E. Adar. How to search a social network. *Social Networks*, 27(3):187–203, 2005.

[5] E. Agichtein and L. Gravano. Extracting relations from large plain-text collections. In *Proc. of the 5th ACM International Conference on Digital Libraries (ACMDL00)*, pages 85–94, 2000.

[6] B. Alema-Meza, M. Nagaraja, C. Ramakrishnan, A. Sheth, I. Arpinar, L. Ding, P. Kolari, A. Joshi, and T. Finin. Semantic analytics on social networks: Experiences in addressing the problem of conflict of interest detection. In *Proc. WWW*, 2006.

[7] K. Balog, L. Azzopardi, and M. Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of SIGIR'06*, 2006.

[8] V. Boer, M. Someren, and B. Wielinga. Extracting instances of relations from web documents using redundancy. In *Proc. the 3rd European Semantic Web Conference (ESWC)*, 2006.

[9] D. Brickley and L. Miller. Foaf vocabulary specification. namespace document. 2005.

[10] S. Brin. Extracting patterns and relations from the world wide web. In *Proc. the WebDB Workshop at 6th International Conference on Extending Database Technology (EDBT)*, 1998.

[11] P. Buitelaar, P. Cimiano, and B. Magnini. *Ontology Learning from Text: Methods, Evaluation and Applications*. IOS Press, Amsterdam, 2005.

[12] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. In *Proc. of ACL '89, Association of Computational Linguistics*, pages 76–83, 1989.

[13] P. Cimiano. Ontology learning and populations. In *Proc. the Dagstuhl Seminar Machne Learning for the Semantic Web*, 2005.

[14] P. Cimiano, G. Ladwig, and S. Staab. Gimme' the context: Context-driven automatic. semantic annotation with c-pankow. In *Proceednings of the 14th International Word Wide Web Conference (WWW)*, 2005.

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

[16] P. Cimiano, S.Handschuh, and S. Staab. Towards the self-annotating web. In *Proc. of the 13th World Wide Web Conference*, 2004.

[17] P. Cimiano and J. Volker. Towards large-scale open-domain and ontology-based named entity classification. In *Proc. RANLP*, 2005.

[18] F. Ciravegna, S. Chapman, A. Dingli, and Y. Wilks. Learning to harvest information for the semantic web. In *Proceednings of the 1st European Semantic Web Symposium*, 2004.

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

[20] A. Culotta and J. Sorensen. Dependency tree kernel for relation extraction. In *Proc. ACL*, 2004.

[21] I. Davis and E. V. Jr. In *RELATIONSHIP: A vocabulary for describing relationships between people, http://vocab.org/relationship/*, 2004.

[22] L. Ding, L. Zhou, T. Finin, and A. Joshi. How the semantic web is being used an analysis of foaf documents. In *Proc. of the 38th Ann. Hawaii International Conference System Sciences*, 2005.

[23] E. T. Dunning. Accurate methods for the statics of surprise and coincidence. 19(1):61–74, 1993.

[24] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Web-scale information extraction in knowitall(preliminary results). In *Proceednings of the 13th International Word Wide Web Conference (WWW)*, 2004.

[25] C. Faloutsos, K. S. McCurley, and A. Tomkins. Fast discovery of connection subgraphs. In *Proc. the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[26] L. C. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1:215–239, 1979.

[27] L. C. Freeman. Visualizing social networks. *Journal of Social Structure*, 1(1), 2000.

[28] G. Geleijnse and J. Korst. Automatic ontology population by googling. In *Proc. BNAIC*, 2005.

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

[30] G. Grefenstette. *Explorations in Automatic Thesaurus Construction*. Kluwer, 1994.

[31] M. Harada, S. Sato, and K. Kazama. Finding authoritative people from the web. In *Proc. JCDL*, 2004.

[32] Z. Harris. *Mathematical Structures of Language*. Wiley, 1968.

[33] T. Hasegawa, S. Sekine, and R. Grishman. Discovering relations among named entities from large corpora. In *Proc. of ACL*, 2004.

[34] T. Hope, M. Hamasaki, Y. Matsuo, Y. Nakamura, N. Fujimura, and T. Nishimura. Doing community: Co-construction of meaning and use with interactive information kiosks. In *Proceedings of the 8th International Conference on Ubiquitous Computing (UbiComp2006)*, pages 387–403, 2006.

[35] N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proc. ACL*, 2004.

[36] R. Kannan, S. Vempala, and A. Vetta. On clustering: Good, bad and spectral. *Computer Science*, 2000.

[37] H. Kautz, B. Selman, and M. Shah. The hidden web. *AI Magazine*, 18(2):27–36, 1997.

[38] H. Kautz, B. Selman, and M. Shah. The hidden web. *AI Magazine*, 18(2):27–36, 1997.

[39] M. Kavalec, A. Maedche, and V. Svatek. Discovery of lexical entries for non-taxonomic relations in ontology learning. In *Van Emde Boas, P., Pokorny, J.,Bielikova, M.,Stuller, J. (eds.). SOFSEM 2004*, 2004.

[40] D. Lin. Automatic retrieval and clustering of similar words. In *Proc. COLING-ACL98*, 1998.

[41] A. Maedche. *Ontology Learning for the Semantic Web*. Kluwer, 2002.

[42] Y. Matsuo, M. Hamasaki, J. Mori, H. Takeda, and K. Hasida. Ontological consideration on human relationship vocabulary for foaf. In *Proc. of the 1st Workshop on Friend of a Friend, Social Networking and Semantic Web*, 2004.

[43] Y. Matsuo, M. Hamasaki, H. Takeda, J. Mori, B. Danushka, H. Nakamura, T. Nishimura, K. Hashida, and M. Ishizuka. Spinning multiple social network for semantic web. In *Proc. AAAI*, 2006.

[44] Y. Matsuo, J. Mori, M. Hamasaki, K. Ishida, T. Nishimura, H. Takeda, K. Hashida, and M. Ishizuka. Polyphonet: An advanced social network extraction system. In *Proc. WWW*, 2006.

[45] Y. Matsuo, H. Tomobe, K. Hasida, and M. Ishizuka. Mining social network of conference participants from the web. In *Proc. of the International Conference on Web Intelligence*, 2003.

[46] A. McCallum, A. Corrada-Emmanuel, and X. Wang. Topic and role discovery in social networks. In *Proc. IJCAI*, 2005.

[47] D. W. McDonald and M. S. Ackerman. Expertise recommender: A flexible recommendation architecture. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'00*, pages 231–240, 2000.

[48] P. Mika. Bootstrapping the foaf-web: an experiment in social networking network mining. In *Proc. of 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web*, 2004.

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

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

[51] S. Milgram. Small-world problem. *Psychology Today*, 1(1):61–67, 1967.

[52] J. Mori, Y. Matsuo, and M. Ishizuka. Finding user semantics on the web using word co-occurrence information. In *Proc. PerSWeb05 Workshop on Personalization on the Semantic Web in 10th Int'l Conference on User Modeling (UM-05)*, 2005.

[53] J. Mori, Y. Matsuo, and M. Ishizuka. Personal keyword extraction from the web. *Journal of Japanese Society for Artificial Intelligence*, 20(5):337–345, 2005.

[54] J. Mori, Y. Matsuo, and M. Ishizuka. Web mining approach for a user-centered semantic web. In *Proc. Int'l Workshop on User Aspects on the Semantic Web in 2nd European Semantic Web Conference (ESWC2005)*, 2005.

[55] J. Mori, Y. Matsuo, and M. Ishizuka. Extracting keyphrases to represent relations in social networks from web. In *Proc. 20th Int'l Joint Conf. on Artificial Intelligence (IJCAI2007)*, 2007.

[56] J. Mori, Y. Matsuo, M. Ishizuka, and B. Faltings. Keyword extraction from the web for foaf metadata. In *Proc. 1st Int'l Workshop on Friend of a Friend, Social Networking and the Semantic Web*, 2004.

[57] J. Mori, Y. Matsuo, M. Ishizuka, and B. Faltings. Keyword extraction from the web for person metadata annotation. In *ISWC Workshop Notes VIII (W8) 4th Int'l Workshop on Knowledge Markup and Semantic Annotation (Semannot2004)*, 2004.

[58] J. Mori, T. Sugiyama, Y. Matsuo, and M. Ishizuka. Real-world oriented information sharing using social network. In *Proc. ACM Group 2005 Conference*, 2005.

[59] J. Mori, T. Sugiyama, Y. Matsuo, H. Tomobe, and M. Ishizuka. Real-world oriented information sharing using social network. In *Proc. 25th Int'l Conf on Social Network Sunbelt*, 2005.

[60] J. Mori, T. Tsujishita, Y. Matsuo, and M. Ishizuka. Extracting relations in social networks from the web using similarity between collective contexts. In *Proc. 5th Int'l Semantic Web Conference (ISWC2006)*, 2006.

[61] V. Raghavan and S. Wong. A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Retrieval*, 35(5), 1998.

[62] A. Schutz and P. Buitelaar. Relext: A tool for relation extraction from text in ontology extension. In *Proc. ISWC*, 2005.

[63] H. Schutze. Automatic word sense dicrimination. *Computational Linguistics*, 24(1), 1998.

[64] J. Scott. *Social Network Analysis: A Handbook*. Sage Publications, London, 2000.

[65] M. A. Smith and A. T. Fiore. Visualization components for persistent conversations. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'01)*, pages 136–143, 2001.

[66] L. A. Streeter and K. E. Lochbaum. Who knows: A system based on automatic representation of semantic structure. In *RIAO*, pages 380–388, 1988.

[67] L. G. Terveen and D. W. McDonald. Social matching: A framework and research agenda. *ACM Transactions on Computer-Human Interaction*, 12(3):401–434, 2005.

[68] P. Turney. Measuring semantic similarity by latent relational analysis. In *Proc. IJCAI*, 2005.

[69] P. Velardi, R. Navigli, A. Cuchiarelli, and F. Neri. Evaluation of ontolearn, a methodology for automatic population of domain ontologies. In *P. Cimiano, and B. Magnini, editors, Ontology Learning from Text: Methods, Applications and Evaluation. IOS Press*, 2005.

[70] B. Yu and M. P. Singh. Searching social networks. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 65–72, 2003.

[71] D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *Machine Learning Research*, 2003(2), 2003.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[89] 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.

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

[91] 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.

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

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

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

[95] 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.

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

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

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

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

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

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

[102] 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.

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

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

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

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

[107] 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.

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

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

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

[111] 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.

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

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

[114] 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*, pages 80–93, 2005.

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

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

[117] S. Wasserman and K. Faust. *Social networkanalysis. Methods and Applications.* Cambridge University Press, Cambridge, 1994.

[118] H. Alani et al. Automatic Extraction of Knowledge from Web Documents. *In Workshop of Human Language Technology for the Semantic Web and Web Services, 2nd International Semantic Web Conference*, Sanibel Island, Florida, USA, 2003.

[119] T. Berners-Lee, J. Hender, O. Lassila. The Semantic Web. Scientic American, 2001.

[120] D. Brickley and L. Miller. FOAF: the 'friend of a friend' vocabulary. http://xmlns. com/foaf/0.1/, 2004.

[121] K. W. Church and P. Hanks. Word Association Norms, Mutual Information, and Lexicography, *Proc. of ACL '89, Association of Computational Linguistics*, 1989.

[122] I. Davis and E. Vitiello Jr. RELATIONSHIP: A vocabulary for describing relationships between people, http://vocab.org/relationship/, 2004.

[123] L. Ding, L. Zhou, T. Finin, A. Joshi. How the Semantic Web Is Being Used An Analysis of FOAF Documents. *In Proc. of the 38th Ann. Hawaii International Conference System Sciences*, 2005.

[124] A. Dingli, F. Ciravegna, D. Guthrie, Y. Wilks. Mining Web Sites Using Unsupervised Adaptive Information Extraction. *In Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary, 2003.

[125] E. T. Dunning. Accurate methods for the statics of surprise and coincidence, *Computational Linguistics*, vol.19, No.1, pp.61–74, 1993.

[126] H. Kautz, B. Selman and M. Shah. The Hidden Web. *AI Magazine*, Vo.18, No.2, pp.27–36, 1997.

[127] J. Mori, Y. Matsuo, M. Ishizuka, and B. Faltings. Keyword Extraction from the Web for FOAF Metadata. *In Proceedings of the 1st Workshop on Friend of a Friend, Social Networking and Semantic Web*, 2004.

[128] Y. Matsuo, M. Hamasaki, J. Mori, H. Takeda and K. Hasida. Ontological Consideration on Human Relationship Vocabulary for FOAF. *In Proceedings of the 1st Workshop on Friend of a Friend, Social Networking and Semantic Web*, 2004.

[129] Yutaka Matsuo, Hironori Tomobe, Koiti Hasida, Mitsuru Ishizuka. Mining Social Network of Conference Participants from the Web. *In Proceedings of the International Conference on Web Intelligence*, pp.190–194, 2003.

[130] , Peter Mika, Bootstrapping the FOAF-Web: an experiment in social networking network mining, *Proc. of 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web*, 2004.

[131] R. Yangarber and R. Grishman. Machine Learning of Extraction Patterns from Unanotated Corpora: Position Statement. *Workshop Machine Learning for Information Extraction*, IOS Press, Amsterdam, pp.76–83, 2000.

[132] Resource Description Framework(RDF) Schema Specification. In *W3C Recommendation*, 2000.

[133] Representing vCard Objects in RDF/XML. http://www.w3.org/TR/2001/NOTE-vcard-rdf-20010222/

[134] Golbeck, J., Hendler, J.: Accuracy of metrics for inferring trust and reputation in semantic web-based social networks. In: Proceedings of the 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW). (2004)

[135] Mika, P.: Ontologies are us: A unified model of social networks and semantics. In: Proceedings of the 4th International Semantic Web Conference (ISWC). (2005)

[136] Brickley, D., Miller, L.: Foaf vocabulary specification. namespace document. (2005)

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

[138] Matsuo, Y., Mori, J., Hamasaki, M., Ishida, K., Nishimura, T., Takeda, H., Hashida, K., Ishizuka, M.: Polyphonet: An advanced social network extraction system. In: Proceednings of the 15th International Word Wide Web Conference (WWW). (2006)

[139] Kautz, H., Selman, B., Shah, M.: The hidden web. AI Magazine **18**(2) (1997) 27–36

[140] Adamic, L.A., Adar, E.: Friends and neighbors on the web. Social Networks **23**(3) (2003)

[141] Harada, M., Sato, S., Kazama, K.: Finding authoritative people from the web. In: Proceedings of the Joint Conference on Digital Libraries (JCDL). (2004)

[142] Culotta, A., Bekkerman, R., McCallum, A.: Extracting social networks and contact information from email and the web. In: Proceedings of the 1st Conference on Email and Anti-Spam (CEAS). (2004)

[143] Scott, J.: Social Network Analysis: A Handbook. Sage Publications, London (2000)

[144] Wasserman, S., Faust, K.: Social network analysis. Methods and Applications. Cambridge University Press, Cambridge (1994)

[145] Grefenstette, G.: Explorations in Automatic Thesaurus Construction. Kluwer (1994)

[146] Lin, D.: Automatic retrieval and clustering of similar words. In: Proceedings of COLING-ACL98. (1998)

[147] Schutze, H.: Automatic word sense dicrimination. Computational Linguistics **24**(1) (1998)

[148] Harris, Z.: Mathematical Structures of Language. Wiley (1968)

[149] Raghavan, V., Wong, S.: A critical analysis of vector space model for information retrieval. Journal of the American Society for Information Retrieval **35**(5) (1998)

[150] Kannan, R., Vempala, S., Vetta, A.: On clustering: Good, bad and spectral. Computer Science (2000)

[151] Zelenko, D., Aone, C., Richardella, A.: Kernel methods for relation extraction. Machine Learning Research **2003**(2) (2003)

[152] Culotta, A., Sorensen, J.: Dependency tree kernel for relation extraction. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL). (2004)

[153] Kambhatla, N.: Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In: Proceedings of ACL. (2004)

[154] Brin, S.: Extracting patterns and relations from the world wide web. In: Proceedings of the WebDB Workshop at 6th International Conference on Extending Database Technology (EDBT). (1998)

[155] Agichtein, E., Gravano, L.: Extracting relations from large plain-text collections. In: Proc. of the 5th ACM International Conference on Digital Libraries (ACMDL00). (2000) 85–94

[156] Buitelaar, P., Cimiano, P., Magnini, B.: Ontology Learning from Text: Methods, Evaluation and Applications. IOS Press, Amsterdam (2005)

[157] Maedche, A.: Ontology Learning for the Semantic Web. Kluwer (2002)

[158] Cimiano, P.: Ontology learning and populations. In: Proceedings of the Dagstuhl Seminar Machne Learning for the Semantic Web. (2005)

[159] Velardi, P., Navigli, R., Cuchiarelli, A., Neri, F.: Evaluation of ontolearn, a methodology for automatic population of domain ontologies. In: P. Cimiano, and B. Magnini, editors, Ontology Learning from Text: Methods, Applications and Evaluation. IOS Press. (2005)

[160] Geleijnse, G., Korst, J.: Automatic ontology population by googling. In: Proceedings of the 17th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC). (2005)

[161] Ciravegna, F., Chapman, S., Dingli, A., Wilks, Y.: Learning to harvest information for the semantic web. In: Proceednings of the 1st European Semantic Web Symposium. (2004)

[162] Kavalec, M., Maedche, A., Svatek, V.: Discovery of lexical entries for non-taxonomic relations in ontology learning. In: Van Emde Boas, P., Pokorny, J.,Bielikova, M.,Stuller, J. (eds.). SOFSEM 2004. (2004)

[163] Cimiano, P., Volker, J.: Towards large-scale open-domain and ontology-based named entity classification. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP). (2005)

[164] Schutz, A., P.Buitelaar: Relext: A tool for relation extraction from text in ontology extension. In: Proceedings of the 4th International Semantic Web Conference (ISWC). (2005)

[165] Cimiano, P., Ladwig, G., Staab, S.: Gimme' the context: Context-driven automatic. semantic annotation with c-pankow. In: Proceednings of the 14th International Word Wide Web Conference (WWW). (2005)

[166] Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D., Yates, A.: Web-scale information extraction in knowitall(preliminary results). In: Proceednings of the 13th International Word Wide Web Conference (WWW). (2004)

[167] Boer, V., Someren, M., Wielinga, B.: Extracting instances of relations from web documents using redundancy. In: Proceedings of the 3rd European Semantic Web Conference (ESWC). (2006)

[168] Matsuo, Y., Hamasaki, M., Takeda, H., Mori, J., Danushka, B., Nakamura, H., Nishimura, T., Hashida, K., Ishizuka, M.: Spinning multiple social network for semantic web. In: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI). (2006)

[169] D. Brickley and L. Miller. FOAF: the 'friend of a friend' vocabulary. http://xmlns. com/foaf/0.1/, 2004.

[170] L. C. Freeman. Centrality in social networks: Conceptual clarification, *Social Networks*, Vol.1, pp.215–239, 1979.

[171] J. Goecks and E. D. Mynatt. Leveraging Social Networks for Information Sharing *In Proc. of CSCW'04*, 2004.

[172] J. Golbeck, J. Hendler, and B. Parsia. Trust networks on the semantic web, *in Proc. WWW 2003*, 2003.

[173] M. Granovetter. Strength of weak ties, *American Journal of Sociology*, Vol.18, pp.1360–1380, 1973.

[174] C. Masolo, S. Borgo, A. Gangemi, N. Guarinno, and A. Oltramari. WonderWeb Deliverable D18, http://wonderweb.semanticweb.org/deliverable/D18.shtml

[175] Y. Matsuo, M. Hamasaki, J. Mori, H. Takeda and K. Hasida. Ontological Consideration on Human Relationship Vocabulary for FOAF. *In Proc. of the 1st Workshop on Friend of a Friend, Social Networking and Semantic Web*, 2004.

[176] Y. Matsuo, H. Tomobe, K. Hasida, M. Ishizuka. Finding Social Network for Trust Calculation. *In Proc. of 16th European Conference on Artificial Intelligence*, 2004.

[177] D. M. Bikel, R. L. Schwartz, and R. M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, Vol.34, pp.211–231, 1999.

[178] F. Keller and M. Lapata. Using the web to obtain frequencies for unseen bigrams. *Computational Linguistics*, Vol.29, No.3, pp.459–484, 2003.

[179] S. Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments, & Computers*, Vol.23, No.2, pp.229–236, 1991.

[180] E. Terra and C.L.A Clarke. Frequency estimates for statistical world similarity measures. *In Proc. of the NAAC/HLT*, pp.165–172, 2003.