ABSTRACT

BATALAGUNDU VISWANATHAN, ARVIND. Bootstrapping Referral Systems With Social Network Information. (Under the direction of Dr. Munindar P. Singh).

This thesis addresses the challenge of facilitating human interactions in solving problems. To this end, it assigns an agent to each user, and models a social network as a multiagent system. A user's agent helps them by sending out and responding to queries on their behalf. Each agent makes its decisions based on its models of the expertise and trustworthiness of other agents. However, such models are not trivial to construct and maintain. This thesis develops an approach wherein the models are seeded based upon information extracted from the user's emails and from existing social networking sites. The main contribution of this thesis is in the specification of heuristics by which expertise and trustworthiness can be computed. It also provides a general schema and methodology by which additional sources of social information can be incorporated.

Bootstrapping Referral Systems With Social Network Information

by

Arvind Batalagundu Viswanathan

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Computer Science

Raleigh, North Carolina

2007

Approved By:

Dr. Laurie Williams

Dr. Edward Gehringer

Dr. Munindar P. Singh
Chair of Advisory Committee

Biography

Arvind Batalagundu Viswanathan was born on the 12th of March,1983, in Coimbatore, India. He got his Bachelor of Technology degree, majoring in Computer Science, from Kumaraguru College of Technology, TN, India. He is currently a graduate student in the Department of Computer Science, North Carolina State University.

Acknowledgments

I would like to thank my adviser, Dr. Munindar Singh, for his guidance, Dr. Laurie Williams and Dr. Edward Gehringer for agreeing to serve on my committee, and my parents, for everything else.

Contents

List of Figures

Chapter 1

Introduction

Seeking the help of friends or other contacts is an approach widely used by humans to solve problems. The friends in turn either provide the answers or refers the person who is seeking help to other contacts who might know the answer. Thus people generally leverage their social-network in getting things done. In computational settings, this interpersonal communication can be modeled through a system in which the entities refer each other based on some criteria. Such systems are called *referral systems*. The referral systems also rely on the social-network information to provide referrals. Constructing a user's social model from scratch involves extracting information from various sources. Many existing referral systems have tried to bootstrap their social models, but the social models constructed are not accurate enough. The social models that these referral systems construct from various information sources do not accurately reflect the underlying real-world social network of an individual. In our thesis, we have come up with a way to model social information from social networking sites and email folders. We have developed a schema specifying the requirement for an information source and procedures to extract information from these information sources.

1.1 Social Networks

A social-network describes how people are related to each other in the real-world. Based on the relationship considered, the structure of social-networks may differ. Many approaches have been suggested to construct such a social model. Here we are interested in constructing a social model from online sources. Next we discuss various sources on the web that could be used for extracting social information.

1.1.1 Sources for Extracting Social Network Information

Social information about a user can be extracted online by visiting the user's web pages [Kautz et al., 1997] or by conducting a web search. Another approach is to search for the co-occurrences of names in papers published [Yu and Singh, 1999]. By parsing a machine-processable format of user profiles, we can extract social information. The contents of email can also be used to construct a social model. Social networking sites, which are each a centralized repository containing user profiles can also be used for constructing social models.

1.1.2 Representation of Social Network

Once the information about the user's social model is captured, we have to represent this information suitably. A social-network can be represented as a graph. In the graph, nodes represent individuals and links between the nodes represent the relation between the individuals. Labels on the links are used to specify additional information about entities. In a social-network graph, the nodes that are directly connected to a given node are called its neighbors.

1.2 Multiagent Referral System

Referral systems are used to model interpersonal communication that happens in the real world. In a referral system, each participating entity is understood as a software module that performs a task on behalf of user. Such software modules are called *agents*. The agents work together to perform their user's task. For our purposes, since an agent has to contact or refer to another agent which it considers an expert, the agent should maintain models about other agents. To model information about other agents, it has to make use of the underlying social-network.

1.3 Motivation

For referral systems to provide accurate referrals, its member agents should capture the user's real-world relationships with accuracy. In this thesis we address how to bootstrap a referral system. In order to refer a neighboring agent, the given agent must know the following information about each neighbors:

- Domain Expertise
- Trust

Domain Expertise

The ability to provide answers to a questions in a particular domain is called domain expertise. Each user modeled as an agent might have different areas of expertise. Based on the level of accuracy, the expertise level of an agent is measured.

Trust

Trust can capture many aspects of a relationship. But typically it would encompass the responsiveness of an agent, accuracy in providing referrals, and so on.

The following real-world example would help us understand the importance of the above attributes in a referral system

A student has some confusion regarding the courses to pursue in a semester. The student, in order to clarify his understanding, first short-lists the different sources of information he can possibly contact. Then he has to decide who in his list of contacts would be able to help. To make this decision, he should know the expertise areas of all his contacts and how forthcoming they would be in providing an answer. The student is able to extract the above information from his social network. When the contact receives a question from the student, he tries to answer the question himself. If he is not able to answer the question himself, he tries to identify a friend who would be able to provide the answer. He then recommends that friend to the student. In the above process the user's social network supports decision making.

In Figure 1.1, we explain how the referral agent is bootstrapped and how the agent makes use of the social information in providing referrals. In our thesis, we capture the domain expertise information of all the neighbors and the model we propose for doing so is common for all information sources.

1.4 Challenges

There have been many efforts to model social network in the past. Some sources that could be used for modeling social- network information were identified. The Referral Web [Kautz et al., 1997] tries to model the social-network information from email content and by visiting the home pages of the individuals. In one of its usage scenarios MARS [Yu and Singh, 1999] infers the social-network information by checking for co-occurences of names in papers. But most of the information sources do not reveal all the information we need. To model the social network accurately we have to choose an appropriate source. The other challenge is that, we have to come up with ways to measure the levels expertise for each user and the approach should be uniform across many information sources. Another challenge we face is in representing the captured information. The representation we choose should be machine processable, hence we have to represent the resulting graph information in Section 1.1.2 in a different format, without loss of data. In the next section we discuss the approach used to bootstrap the referral system.

1.5 Approach

We have specified a schema to identify the information sources that could be used to bootstrap our system. The techniques we use to extract the domain expertise are uniform across the different information sources. By applying transformations on the different sources of information, we can convert them to a format that is common for all the information sources. From this format, we can identify the expertise information of users in social network. This format also serves as a framework from which the trust information can be inferred. The system can be understood by referring to 1.1. Once we identify a source of information, we perform the following steps to extract the social-network information:

- Develop a Common Schema

The information that is required to capture the social-network information is specified as a schema. The techniques to capture the domain expertise and trust can be applied when the information is converted to a format specified in the schema.

- Apply a transformation

  In most cases the information might not directly resemble the format specified in schema. To make it resemble the format specified in the schema, we apply transformations.

- Extract Social-network Information

  This is done by identifying a user's friends in our information source. We extract the domain expertise of each user by mining the content of the information source. This is a common approach; hence it may not measure the expertise information as accurately as some of the information source specific extraction techniques developed [Yu and Singh, 1999].

- Represent the Social-network Information

  The social-network information identified has to be represented in a format that agents can process. This representation is used for reference whenever the agent has to provide a referral.

1.6 Thesis Organization

This thesis studies the process of bootstrapping a referral system by constructing a social network. The subsequent chapters describe how we achieve this in the following order: Chapter 2 presents the architecture of our system and the algorithms we have used for extracting domain expertise information. Chapter 3 discusses the implementation of the system and chapter 4 discusses about some of the existing work in this domain and extensions that could be added.

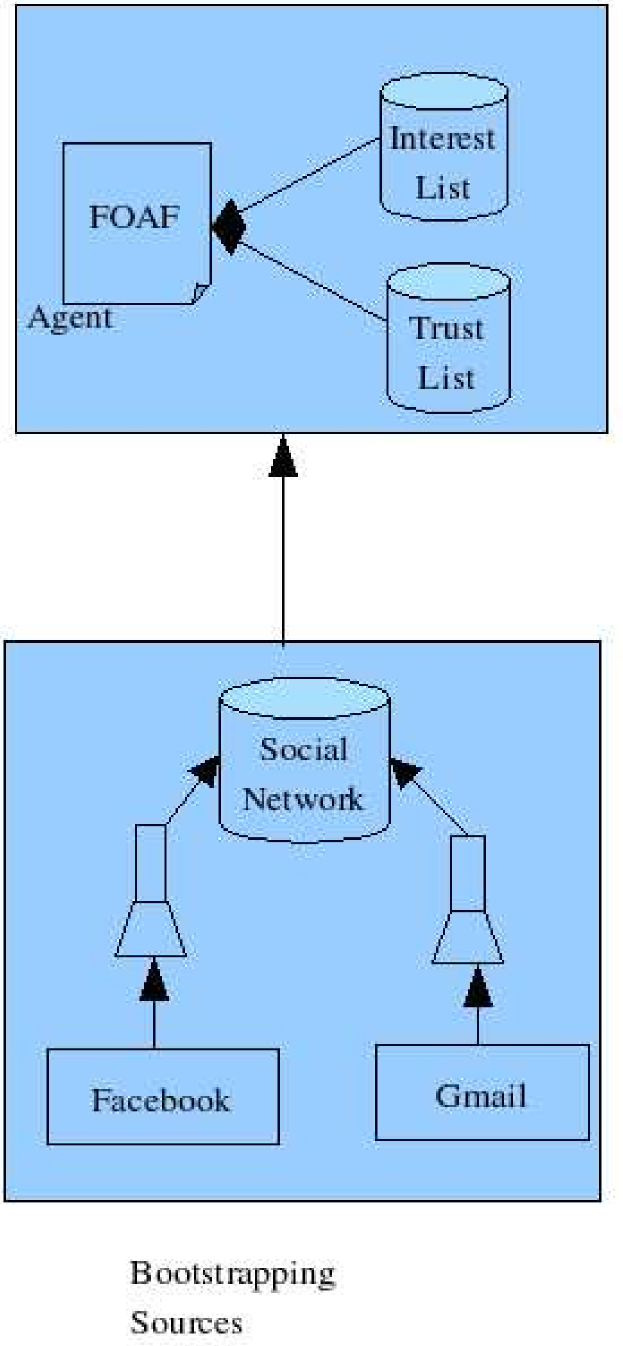

Figure 1.1: Bootstrapping process

Chapter 2

Architecture

Many information sources could be used to extract social information. Our goal is to come up with a common approach to extract social information from different information sources. The architecture of our system is shown in Figure 2.1.

2.1 Overview

The information from the sources is transformed to a format specified in the schema. This is handled by the adapter module. The next step is to extract the social models from these sources. This extraction technique is common across all sources since it is applied on the information represented in the format specified by the schema. The content of the sources is mined to identify the areas of expertise of all the neighbors. The technique to extract the areas of expertise and to measure the expertise levels are described later in the chapter. Finally the extracted information is represented in a format which can be processed by agents. This representation should also accommodate the information extracted from different sources.

2.1.1 Information Sources

Online sources where we can find information about people are personal home pages, online community, corporate logs, and so on. To determine the closeness between people, existing analysis techniques determine frequency of co-occurrence of names. For example, co-authors of all papers written by A are considered as friends of A and the number of times these people have co-authored with A determine the closeness. To identify the interests of a person, term frequency inverse document frequency (TF-IDF) is applied. TF-IDF helps in measuring the importance of a term in a document. This measure helps extracting the context of a document. This is the approach adopted by [Kautz et al., 1997] to capture social information. But the sources used in ReferralWeb [Kautz et al., 1997] to capture social information only consider researchers. Having a centralized site storing this information would help us focus on the analysis part rather than the data collection part. In such a central site, the information we are interested in is explicit. In our system we make use of Facebook and Gmail as sources.

2.1.2 Social Networking Sites

Social-networking sites are collections of online profiles. The profile information typically includes an individual's list of friends, interests, messages exchanged. Another interesting aspect of a social network is the existence of communities. Similar to the way in which communities are created in real life, an individual in a social network creates a community and invites people who he thinks have similar interests. Communities are based on specific areas of interest.

Some important definitions follow:

Definition 1 *Betweenness* is how often the individual is used as a bridge to connect people. We show later how betweenness is used to determine the trust value of different users.

Definition 2 *Centrality* is the measure of number of friends a person has. In a social network, the number of links leaving the node denoting an individual is the centrality of a person.

Definition 3 *Clustering coefficient* is a measure of how interconnected the neighbors of a node are. It is the ratio of number of links connecting the nodes in the neighborhood to the number of all possible links that can exist between the nodes in a network. This measure is used to categorize the network, whether it is regular, small world, or random. Regular graphs 2.2 have a high clustering coefficient. Random graph has low clustering coefficient 2.2 . Small world graphs have an intermediate clustering coefficient value 2.2.

Table 2.1: Population of social networking sites in April-2006. Information from Nielson Rating.

site name	Population
MySpace.com	51441
Classmates.com	14792
Facebook.com	14069
Youtube.com	12669
MSN Spaces	9566
Xanga.com	7146
Flickr.com	5163
Yahoo! 360	4936

Definition 4 *Path length* is the number of nodes between the source and target node.

Table 2.2: Growth of social networking sites. Population in Mar-2007. Information from Nielson Rating.

site name	Population
MySpace.com	172,296,430
Classmates.com	30,000,000
Facebook.com	55,000,000
Youtube.com	12,500,000

Background

Several analyses have been carried out on network structures, as their characteristics reveal interesting information. The working of a social network can be attributed to the small-world phenomenon. According to the small-world hypothesis, everyone in the world can be reached through a short chain of social acquaintances [Milgram, 1967]. The psychologist Stanley Milgram conducted a research to find out how many referrals he may have to pass through in order to connect two random individuals in the US. He found out that, on average, to connect two individuals we have to pass through six referrals. Table 2.1 gives a good idea of the population of some popular social networking sites. Table 2.2 discusses the growth rates of these social networking sites. Next we discuss properties which could be inferred from the social networking sites.

Social Capital is the measure of importance of an individual in a community.

Techniques to measure social capital, and to use it to move from a disadvantaged position in a social network to one rich in opportunity are suggested by [O'Connor and Sauer, 1905]. All these decisions are based on the content available in the social network. Social networking sites provide a way to carry out search on the content available. Thus the social networking sites serve the role of social search engines as well.

By tracking the browsing pattern of the users, it is possible to determine their interests [Seo and Zhang, 2000]. Another way to study a user's interest is by analyzing the online documents classified by the user [Krulwich, 1995].

2.1.3 Email

Emails are another source of information we use for extracting the social-network information. Typically email information contains the sender's and the recipient's addresses and the messages exchanged between them. Using this information, we can model the acquaintances and the areas of expertise of each acquaintance. We assume that people exchange emails with contacts they know well. Another assumption we make is that using email logs as a supporting source of information to build social network can make the model more accurate. The main reason for choosing email feeds as a source of information was the format in which email service providers were delivering the content through their API's.

There have been many attempts in the past to mine information from email [Culotta et al., 2004]. But by using email feeds as their only source of information they have not been sufficiently accurate in their analysis. Yet many services have been developed by mining email content and they have proved to be useful. Spam filtering requires the email clients to mine the content of the email. The significance of mining email contents in forensics is explained in [de Vel et al., 2001]. Some challenges have prevented the mining of the emails as a basis for social-network analysis. Not many email service providers offer API. Another important problem in using email as information source is the issue of privacy. Making user-generated content accessible publicly is risky. Specifically the risk involves allowing a third party software to mine the contents of the email. As far as addressing this risk is concerned, email service providers themselves are offering services that mine the contents of the email. Since the email service providers are the ones who mine the content this is considered legal. Gmail mines the user email content extensively and provides targeted advertising, which has proved to be a big hit. Another useful project is a recipient-recommendation system, where based on the content the email client suggests the possible recipients for the message. Many intra-company email clients have been mining the content of the users mail and helping filter the spam.

2.2 Input and Output Schema

A schema specifies the format or structure that documents must follow. In object-oriented terms, a schema can be compared with class and documents that satisfy the schema can be compared with instances. Developing a schema would enable, the procedures that ensue to be applicable to all the information sources. Since we have many information sources, our aim is to represent the information contained in them in a format specified by the schema.

2.2.1 Schema for Input

Before developing an input schema we have to identify the information that the information sources have to contain in order to extract the social network information. In our case we are interested in identifying the acquaintances of a person and areas of interest of the acquaintances. Since emails include the address of a correspondent, the list of acquaintances can be easily extracted. However, the areas of expertise information is generally not explicit in most of the information sources. We have developed techniques that could be used to capture the areas of expertise of users. In our approach, we use message exchanges as sources for extracting the areas of expertise information. Hence our schema should contain the list of acquaintances and the messages exchanged between the user and his acquaintances. This is represented in Figure 2.3

Our schema information only captures the basic information needed to capture social-network information. Some of the social networking sites store these information explicitly and obviate the need for transformation to a schema. In our case the information extracted from social networking site would directly satisfy the requirements mentioned in the schema, but to model more accurately, we make use of the profile information of the user. The information from the input source is transformed to an instance satisfying the schema, before we apply the domain interest extraction techniques. In the implementation section we have captured this using the format using an XML schema.

2.2.2 Schema for Output

The information extracted from social sources can be formally modeled as a graph as mentioned in Section 1.1.2. However, this information cannot be directly processed by agents. We need to represent the information in a format that agents can process. Before specifying how we capture the social information, we give an overview of the information that we extract from all the information sources. We capture the following information in the social network we model:

- List of acquaintances.
- Domain expertise calculation.
- Trust level associated with each acquaintance.

This is the information we capture at the end of our approach. Since the structure of information remains same for all the individuals, we can make use of a schema. Since the information is about an individual, his expertise, his friends and so on, a schema used to describe a person can be extended in our case. This is represented in Figure 2.4 .Such a schema already exists and we make use of it in our model. This schema is based on Resource Description Framework and is discussed in Section 3.3.2.

2.3 Domain Expertise Calculation

The social-network information captured in the format specified by the input schema is then used for extracting expertise information of each user. In the case of social networking sites this information is explicit and our system only tries to measure the levels of expertise. But in the case of emails, the areas of expertise have to be inferred from the message exchanges between the user and his correspondents, besides measuring the levels of expertise. We see in detail the procedures to extract the domains of expertise information and measuring levels of expertise.

2.3.1 Extracting Domain Expertise

Many techniques have been developed to extract domain expertise information of a user from the Internet. In [Krulwich, 1995], an agent lets the user collect the documents he likes. The collected documents are categorized based on the relevant field. From this document set important phrases are extracted. These phrases indicate the user's expertise on the category specified. Based on this, additional documents are added to the set after getting the feedback from the user. A similar strategy is adopted in MARS [Yu and Singh, 1999]. Another approach to identify a user's expertise is by making use of *collaborative filtering*. In collaborative filtering, we first identify users who have similar preferences to the preference of an active individual. Then based on the expertise of the users identified in the previous step, we predict expertise of the active user. Amazon uses this technique to provide recommendations to its users.

The common strategy in the above approaches is extracting the context from the documents or profile information. In case of emails, the context extracted from the email conversations would help in identifying the domains of expertise of individuals. In our case, we identify all the messages that a user and his acquaintance have exchanged. We then extract the important keywords from these messages and consider them as areas of expertise. For social networking sites, even though in most of the cases the social-network information can be extracted directly, we extract the keywords from the expertise areas specified by the user. To understand this situation better consider the following example.

A user in a social networking site mentioned "Did I mention I 'am into computers ?" in his interest field. The underlying expertise area can be inferred by humans but for machines, we extract the keywords from this sentence and use it as a area of expertise. In our case we could apply the keywords for all the information sources, but we restrict ourselves to cases where a area of expertise has more than four words in it. A web service call is used to extract the context information from the message. The details of the approach are discussed in Section 3.4.

2.3.2 Measuring Expertise Level

After extracting the domain expertise of a user, we calculate the level of expertise of the user in each area of expertise specified. In our approach we use the frequency of the occurrence of keywords in a set of documents to determine the expertise level of the user. There are more complex techniques to measure expertise level like TF-IDF, but to preserve the simplicity of implementation we use frequency count In the case of emails, the document refers to the messages exchanged between the user and his acquaintance and in the case of social networking sites the document refers to the description of the groups that a user has subscribed to. The assumption we make is that the user is generally part of a community he likes and consequently he knows more about the community. For example, once we have identified that a friend has an interest in music we measure the expertise by counting the number of exchanges in which he discussed music. This measure is useful when there are many users in an individual's social circle with the same areas of interest. The reason why an individual's friends tend to have same fields of interest can be attributed to the fact that people prefer to have relationship with others who have similar interests [cliques in small world]. This process of determining the level of expertise is not a one-time process. Since the user's level of expertise can change over a period of time, it is possible to change the expertise value based on the responses it provides. An individual can also change the modeled level of expertise of a friend by specifying how satisfied he was with an answer received from the friend. Since this measure varies based on the user, only the user's feedback can make it accurate. But the values we assign during bootstrapping are important because they help avoid conflicts, that may arise when many users in a person's network have the same interest.

2.3.3 Trust Calculation

Trust is another parameter used by agent when deciding what referrals to give. The calculation of trust is important because it determines the quality of responses or referrals. Having a good trust model should help us in making predictions about neighboring agents.

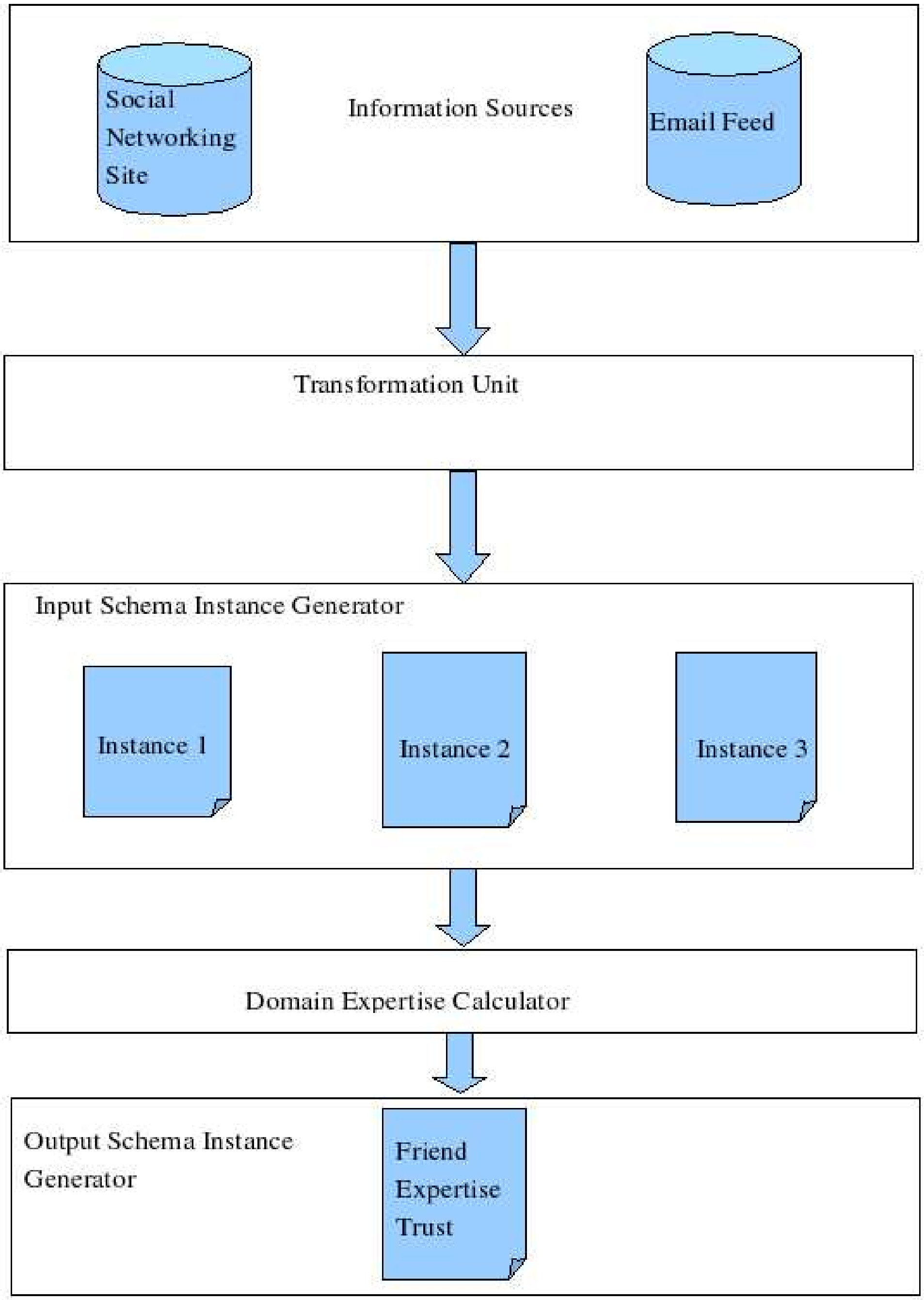

Figure 2.1: Architecture

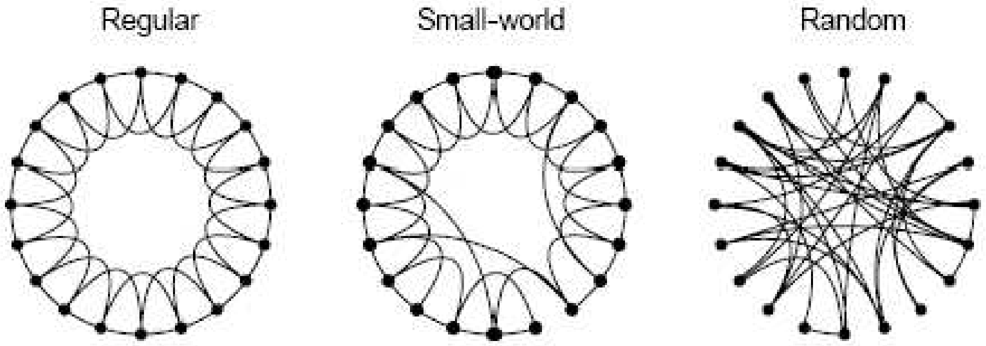

Figure 2.2: Types of Network Based on Watts and Strogatz Model

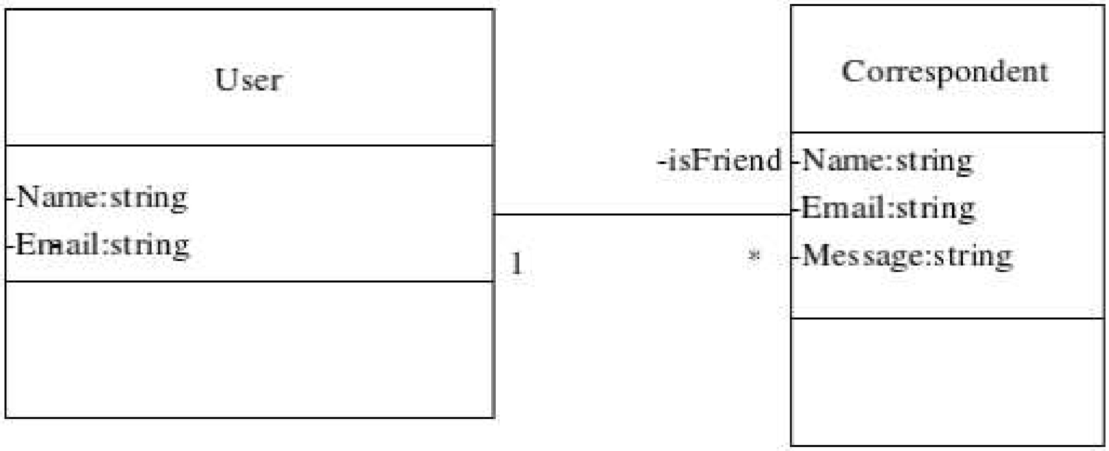

Figure 2.3: Graphical Representation of Input Schema

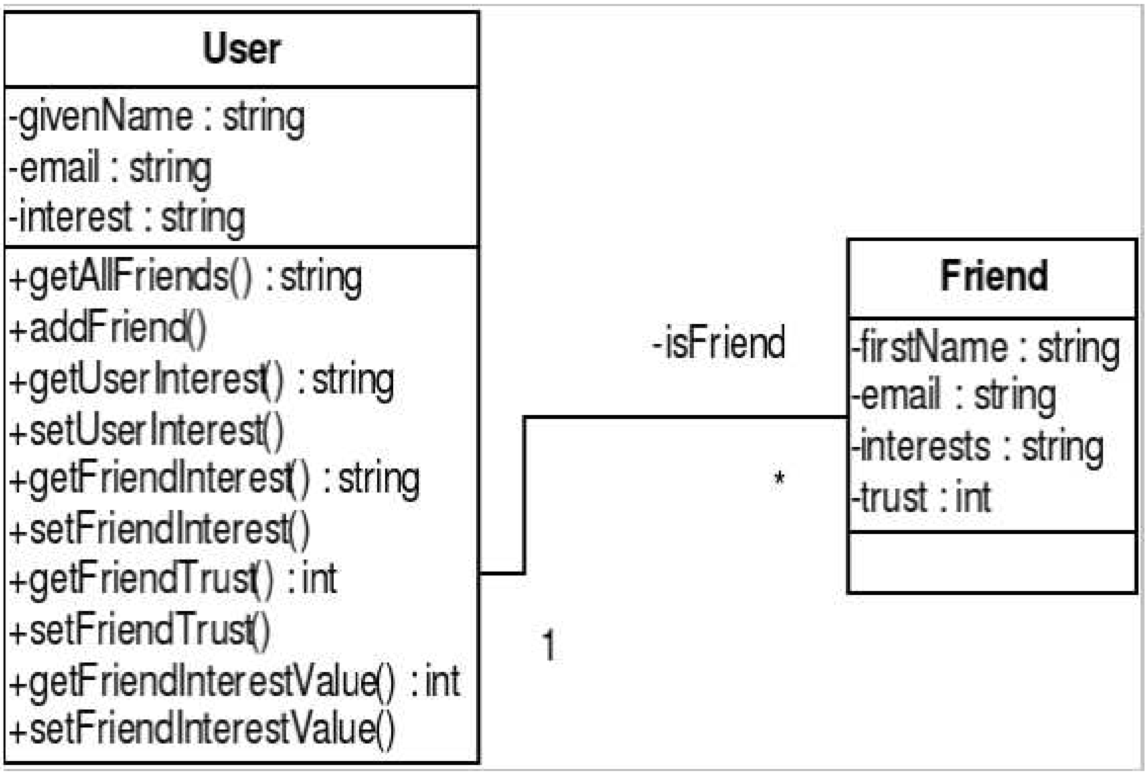

Figure 2.4: Graphical Representation of Output Schema

Chapter 3

Implementation

This chapter describes how we implemented the architecture described in the Chapter 2. We present a short overview of the different modules and the technologies used in implementing them.

Our application is developed using Mozilla's XUL-based framework. The agent communication part is implemented using Jabber [Jabber], which is an open-source instant messaging platform. For modeling the social network we use a representation called FOAF (Friend of a Friend) [FOAF]. FOAF is a vocabulary to capture information about people. It is based on XML/RDF, making it readable by agents. For the network analysis part we have used DOM parsing of XML and XSLT. These are required because the messages extracted from various information sources are in XML.

3.1 Implementation Overview

To get the social information of a user, the contents of many online social-network sources have to be mined. The information sources used for extracting social-network information, typically contain the following information:

- List of acquaintances.
- Email address of each acquaintance.

- Messages exchanged between the user and each of his acquaintance.

Even though most of the information sources, contain the above information, the structure of the information differs from one source to another. Because of this difference in structure a uniform way to extract the social-network data cannot be formulated. To understand the situation, consider the following two information sources in Figure 3.1 and Figure 3.6, from which the user wants to extract social information.

Listing 3.1: Schema for Input

```
<?xml version='1.0' encoding='UTF-8'?>
   <feed version='0.3' >
     <entry>
       <title>[Comeeko] Welcome to Comeeko!</title>
        <summary>Welcome to Comeeko. Your login
         credentials are:
        </summary>
        <author>
          <name>help</name>
          <email>help@comeeko.com</email>
        </author>
    </entry>
    <entry>
      <title>[Comeeko] Welcome to Comeeko!</title>
      <summary>Welcome to Comeeko. Your login credentials are:
      </summary>
      <author>
        <name>help</name>
        <email>help@comeeko.com</email>
      </author>
   </entry>
   <entry>
     <title>[Comeeko] Welcome to Comeeko!</title>
     <summary>DOnt get lost in the woods</summary>
```

```
      <author>
         <name>help</name>
         <email>arvind.viswanathan@gmail.com</email>
      </author>
   </entry>
   <entry>
      <title>[Comeeko] Welcome to Comeeko!</title>
      <summary>Ipod is not the best music player</summary>
      <author>
         <name>help</name>
         <email>govind.viswanathan@gmail.com</email>
      </author>
   </entry>
</feed>
```

In both the cases the information source contains list of acquaintances, email address of each acquaintance, and messages exchanged between the given user and acquaintance, but the structure in which this information is represented differs in both the cases. Our aim in this thesis is to use a common approach to extract social information from all the sources. To achieve this, we have introduced the transformation step before extracting the social information. The purpose of transformation is to map any information source to a format satisfying the schema. To do the mapping, location of the elements in the input source are important. This location information can be specified as an XPath expression. The user provides XPath expressions to map the information in the input source to a format satisfying the schema. Having a transformation module obviates the need for a specialized domain expertise extraction technique for each source. In our application the mapping information for Facebook and Gmail are built in the system. For other information sources, the mapping information has to be specified by the user. For example, the mapping information of the friend list, email, and messages for the information source in Figure 3.6 is Figure 3.8. The output FOAF file generated by extracting the social information from Facebook account of a user is shown in Figure 3.2.

Listing 3.2: Schema for Input

```
<?xml version=”1.0”?>
<RDF:RDF xmlns:NS1=”http://xmlns.com/foaf/0.1/”
         xmlns:NC=”http://home.netscape.com/NC−rdf#”
         xmlns:RDF=”http://www.w3.org/1999/02/22−rdf−syntax−ns#”>
  <NS1:Person RDF:about=”Naren”
              NS1:givenname=”Naren”
              NS1:mbox=”Naren@dickens”
              NS1:interest=”books−8,chess−8,computers−8
              mention−8,movies−8”
              NS1:geekcode=”3” />
  <NS1:Person RDF:about=”Sankar”
              NS1:givenname=”Sankar”
              NS1:mbox=”Sankar@dickens”
              NS1:geekcode=”8” />
  <NS1:Person RDF:about=”Arjun”
              NS1:givenname=”Arjun”
              NS1:mbox=”Arjun@dickens”
              NS1:interest=”driving−5,eating−5,frisbee−6
              ,playing−6,sleeping−5”
              NS1:geekcode=”3” />
  <NS1:Person RDF:about=”Ranjith”
              NS1:givenname=”Ranjith”
              NS1:mbox=”Ranjith@dickens”
              NS1:geekcode=”3” />
  <NS1:Person RDF:about=”Arvind”
              NS1:firstName=”Arvind”
              NS1:mbox=”Arvind@dickens”
              NS1:interest=”Reading , Religion , Football”>
    <NS1:knows RDF:resource=”Ranjith”/>
    <NS1:knows RDF:resource=”Arjun”/>
    <NS1:knows RDF:resource=”Sankar”/>
    <NS1:knows RDF:resource=”Naren”/>
  </NS1:Person>
</RDF:RDF>
```

By adding new input sources to the system, the FOAF file can be modified to include the new social information of the user. For example, by providing Gmail feed as an additional source of information, the FOAF file now contains the user name Adam in the given user's friend list as shown in Figure 3.3.

Listing 3.3: Schema for Input

```
<?xml version=”1.0”?>
<RDF:RDF xmlns:NS1=”http://xmlns.com/foaf/0.1/”
          xmlns:NC=”http://home.netscape.com/NC-rdf#”
          xmlns:RDF=”http://www.w3.org/1999/02/22-rdf-syntax-ns#”>
<NS1:Person RDF:about=”file:///C://Arjun”
              NS1:givenname=”Arjun”
              NS1:mbox=”Arjun@dickens”
              NS1:interest=”driving -5,eating -5,frisbee -6,
              playing -6,sleeping -5”
              NS1:geekcode=”3” />
  <NS1:Person RDF:about=”Adam”
              NS1:givenname=”Adam”
              NS1:mbox=”adam@dickens”
              NS1:interest=”arvind -9,downloads -6,mail -10”
              NS1:geekcode=”2” />
  <NS1:Person RDF:about=”file:///C://Arvind”
              NS1:firstName=”Arvind”
              NS1:mbox=”Arvind@dickens”
              NS1:interest=”Reading , Religion , Football”>
    <NS1:knows RDF:resource=”file:///C://Ranjith”/>
    <NS1:knows RDF:resource=”file:///C://Arjun”/>
    <NS1:knows RDF:resource=”file:///C://Sankar”/>
    <NS1:knows RDF:resource=”file:///C://Naren”/>
  </NS1:Person>
  <NS1:Person RDF:about=”file:///C://Ranjith”
              NS1:givenname=”Ranjith”
              NS1:mbox=”Ranjith@dickens”
              NS1:geekcode=”3” />
  <NS1:Person RDF:about=”file:///C://Naren”
              NS1:givenname=”Naren”
              NS1:mbox=”Naren@dickens”
              NS1:interest=”books−8,chess−8,computers−8,
              mention−8,movies−8”
              NS1:geekcode=”3” />
  <NS1:Person RDF:about=”Arvind”
              NS1:firstName=”Arvind”
              NS1:mbox=”Arvind@dickens”
              NS1:interest=”books, reading”>
  </NS1:Person>
  <NS1:Person RDF:about=”file:///C://Sankar”
              NS1:givenname=”Sankar”
              NS1:mbox=”Sankar@dickens”
              NS1:geekcode=”8” />
</RDF:RDF>
```

3.2 Schema for Input

In Section 2.1.1, we specify the requirements for information sources. Since this information is common to all the sources, we have expressed this information as a schema

3.2.1 XML Schema

The schema defines the structure and the datatypes used therein. XML schema helps to capture the rules as well as datatype constraints. In our case, we are interested only in capturing the structural information. Many languages are available for capturing schema, but in our approach to capture the social information we use XML Schema. The reason for choosing XML Schema is because the API's used for retrieving the social information in the sources we consider use XML for message exchanges. The XML schema capturing this information is shown in Figure 3.4.The output FOAF file generated by extracting the social information from Facebook account of a user is shown in Figure

Listing 3.4: Schema for Input

```
<xsd:schema xmlns:xsd=”http://www.w3.org/2001/XMLSchema”>
<xsd:element name=”User” type=”personType” />
<xsd:element name=”Friends” type=”collectionType” />
<xsd:element name=Friend type=friendType>
<xsd:element name=Friend type=friendType>
<xsd:complexType name=”personType”>
     <xsd:element name=”name” type=”xsd:string” />
     <xsd:element name=email type=xsd:string />
     <xsd:element name=friends type=groupType />

</xsd:complexType>
<xsd:complexType name=”groupType” minOccurs=”1”>
        <xsd:element name=”friend” type=”friendType”
         minOccurs=”1” />
</xsd:complexType>
<xsd:complexType name=friendType>
        <xsd:sequence>
                <xsd:element name=”Name” type=”xsd:string” />
        <xsd:element name=”Email” type=”xsd:string” />
             <xsd:element name=Message type=xsd:string
             minOccurs=1 />
        </xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

Listing 3.5: Instance Satisfying Input Schema

```
<? xml version=”1.0”?>
<user>
    <name>Adam</name>
    <email>adam@jabber.com</email>
    <correspondents>
    <correspondent>

        <name>Amy</name>
        <email>amy@jabber.com</email>
        <message>I am working on Friday</message>
        <message>I bought an ipod from Apple
        stores</message>

    </correspondent>
    <correspondent>
        <name>Arjun</name>
        <email>arjun@jabber.com</email>
        <message>deliver the pizza on monday</message>
        <message>Visiting France on Friday</message>
    </correspondent>
    <correspondent>
        <name>Orkut</name>
        <email>orkut@jabber.com</email>
        <message>attending football match on
        23rd July</message>
        <message>going to library this sunday</message>

    </correspondent>
    </correspondents>
</user>
```

The requirement that all the information sources should contain information about friends is captured by an indicator *minOccurs*. The next items we capture is the attributes of the friend:name and email. In our approach we use name to identify the friend. The reason for not using email address is because email address of the users are not provided by social network because of privacy. The last requirement to capture is the set of messages that the given user has exchanged with all his friends. This information is captured in the message element. An instance document satisfying the schema is shown in Figure 3.5.

The information source often does not contain information in the same structure specified by the schema. In such cases, the user provides XPath information that specifies the location of friends name, email and messages. We apply transformations using these XPath expressions to convert them in a format satisfying the schema.

3.3 Schema for Output

The output social information we model is common for all users. Hence we specify a schema for capturing this information. The requirements for output are discussed in section 2.2.2. Our requirement is to model a schema that can capture the network information of user along with their areas of expertise and trust level for each of the friend.

3.3.1 FOAF

For our purposes, a schema modeling a person in the real world can be readily extended for our representation. Such a schema already exists and is based on RDF.Friend of a Friend is the representation used to describe people in real world. Using a FOAF representation we can capture any information related to a person. FOAF maintains a vocabulary to define information about people. We are interested in FOAF, because its vocabulary supports capturing relationship between people and interest information of people. The underlying framework for FOAF is Resource Description Framework.

3.3.2 Resource Description Framework

RDF is a language in which the resources can be represented in a machine understandable way. The concept of a semantic web is based on the ability to represent the information in the web in a format in which machines can understand the content and take decisions based on it.

In RDF information is captured as a set of triples. Each triple has a subject, object and property connecting the subject and the object. In our application our subject could be a person. Friendship is the property connecting different resources in our case.

Resource Description Framework Schema (RDFS) is the language we use to represent the schema. RDFS has the basic elements for defining ontologies. Here RDFS:Class is used to declare any resource.A resource is the subject of an RDF statement. RDF:Property is the primitive by which we specify the properties of a class. When specifying a property we also specify the domain and range across which it is defined. *Domain* is the class or resource that it is a part of and *range* is the value that this property takes.

Some of the properties of FOAF that we use in our representation are:

- Foaf:person -This is the property we use to denote any person.
- Foaf:name -This property is used to indicate the name of the person.
- Foaf:knows -This property is used to indicate all the contacts an individual has.
- Foaf:interest -This property is used to indicate the interests of the user.

In our representation, when there are multiple areas of interest, we separate them by ',' and include them in the foaf:interest field. The expertise value of each field of interest is appended with the field name.

To parse the RDF files in Mozilla framework, we use a library provided by Mozilla [RDFDS]. Using this library the FOAF document under consideration can be treated as a set of RDF triples as against an XML document.

3.3.3 Transformation

The input from the information source has to be transformed in to a format satisfying our schema. To perform the transformation the user has to specify XPath expressions. For example, consider the email source in 3.6.The user compares the input with the sample instance specified in 3.5. The user then specifies the XPath expressions to locate acquaintance name, messages, and the email address of the acquaintance. The XPath expressions corresponding to the above information are shown in 3.6. Through the XPath expressions, the user input is transformed to the format specified in 3.7.

Listing 3.6: Sample Input for Extracting Social Information

```
<?xml version=”1.0” encoding=”UTF−8”?>
<GetMessageResponse>
<message>
        <mid>1_326196_AIzJjkQAALMcRQlr7QE89VmzZIQ</mid>
        <receivedDate>1158245354</receivedDate>
        <subject>AAdvantage eSummary for Sept 06</subject>
        <from>
           <name>AADVANTAGE</name>
           <email>esummary@aadvantage.info.aa.com</email>
        </from>
        <to>
           <name></name>
           <email>SMITH@YAHOO.COM</email>
        </to>
        <text>The receipt notice for your OPT
         application has arrived in OIS. Please
         come to OIS at your earliest convenience
         to pick up your receipt notice.You may
         follow the instructions on the receipt
         notice to check your application status
         online.
        </text>
</message>
<message>
        <mid>1_326196_AIzJjkQAALMcRQlr7QE89VmzZIQ</mid>
        <receivedDate>1158245354</receivedDate>
        <subject>Hi how are you</subject>
        <from>
           <name>vikram</name>
           <email>vikram@yahoo.com</email>
        </from>
        <to>
           <name></name>
           <email>SMITH@YAHOO.COM</email>
        </to>
        <text>Microsoft Student Partners will be
         holding a Vista/Office information session
         on April 25. All students and faculty are
         invited. I've included a document with all
         the information and was wondering if you
         could sent that information out to all
         CSC students and faculty.
        </text>
</message>
<message>
        <mid>1_326196_AIzJjkQAALMcRQlr7QE89VmzZIQ</mid>
        <receivedDate>1158245354</receivedDate>
        <subject>New Ipod on sale</subject>
        <from>
          <name>Adam</name>
          <email>adam@yahoo.com</email>
        </from>
        <to>
                <name></name>
                <email>SMITH@YAHOO.COM</email>
        </to>
        <text>
        The Libraries Student Assistant Program
        committee will hold Focus Group sessions
        in April to gather information on student
        worker experiences here in the Libraries.
        * All Libraries' student workers are invited
         to attend one of the following sessions:*
        </text>
</message>
<message>
        <mid>1_326196_AIzJjkQAALMcRQlr7QE89VmzZIQ</mid>
        <receivedDate>1158245354</receivedDate>
        <subject>Party tonight</subject>
        <from>
          <name>Raj</name>
          <email>Raj@yahoo.com</email>
        </from>
        <to>
          <name></name>
          <email>SMITH@YAHOO.COM</email>
        </to>
        <text>According to our records, your current
         period of authorized stay in the US will expire
         in 30 DAYS and you need to let us know if you
        will need to do a program extension, need to apply
        for OPT or you will be departing the US. Once
        your end date arrives we will not be able to
        help you. IF YOU HAVE ALREADY DONE ONE OF THE
        BELOW ITEMS,THEN DO NOT RESPOND.
        </text>
</message>
</GetMessageResponse>
```

3.3.4 XPath Expressions

The user should specify the XPath expressions corresponding to the following elements in the instance document specified in FigureXpath:

- XPath for friend: This expression specifies the location of user information in the input XML. This element includes the names, email and messages of the acquaintance.
- XPath for Email: This expression specifies the location of the email address of the acquaintance in the input XML.
- XPath for Name: This expression specifies the location of the name of the acquaintance in the input XML.
- XPath for Messages: This expression specifies the location of the messages of the given user.

Listing 3.7: Output After Transformation

```
 <? xml version=”1.0”?>
<user>
        <name>Smith</name>
        <email>smith@jabber.com</email>
        <correspondents>
        <correspondent>
          <name>AADVANTAGE</name>
          <email>esummary@aadvantage.info.aa.com</email>
          <message>IThe receipt notice for your OPT
           application has arrived in OIS. Please
          come to OIS at your earliest convenience to pick
          up your receipt notice. You may follow the
          instructions on the receipt notice to check
          your application status online.
         </message>

        </correspondent>
        <correspondent>
          <name>vikram</name>
          <email>vikram@yahoo.com</email>
          <message>Microsoft Student Partners will be
```

```
  holding a Vista/Office information session on
  April 25. All students and faculty are
  invited. I've included a document with all
  the information and was wondering if you
  could sent that information out to all CSC
  students and faculty.
  </message>
</correspondent>
<correspondent>
  <name>Adam</name>
  <email>adam@yahoo.com</email>
  <message>The Libraries Student Assistant
  Program committee will hold Focus Group
  sessions in April to gather information on
  student worker experiences here in the Libraries.
  * All Libraries' student workers are invited
  to attend one of the following sessions:*
  </message>

</correspondent>
<correspondent>
  <name>Raj</name>
  <email>raj@yahoo.com</email>
  <message>According to our records, your current
  period of authorized stay in the US will expire
  in 30 DAYS and you need to let us know if you
  will need to do a program extension, need to apply
  for OPT or you will be departing the US.Once your
  end date arrives we will not be able to help you.
  IF YOU HAVE ALREADY DONE ONE OF THE BELOW ITEMS,
  THEN DO NOT   RESPOND.
  </message>
```

```
        </correspondent>
        </correspondents>
</user>
```

Listing 3.8: Xpath for Mapping Input Source

```
Xpath for Friend:

//message

XPath for Email:

//message/from/email

XPath for Friend Name:

//message/from/name

XPath for Message:

//message/text
```

3.4 Domain Expertise Extraction

To extract the expertise information from information sources, we extract the context of the message exchanged. In section 2.3 other approaches have been suggested. Here we infer the keyword information by making a Webservice call. In our case, we use the keyword search Webservice offered by Yahoo to carry out this process. This Webservice takes as an argument the sentence which we want to analyze. The Webservice internally makes use of Keyword Extraction Algorithm (KEA) [tau Yih et al., 2006]. This algorithm extracts the context from the information it is fed.

In KEA, the system maintains a collection of training documents. Each of these training documents also contain the user inferred context with them. On these documents, Term Frequency * Inverse Document Frequency (TF * IDF) is used along with the relative position of each term's first occurrence in the document [Yu and Singh, 1999]. Since this API is public, developers contribute more documents to the training set, making the system more and more accurate. Another advantage of using a Webservice is that we can easily plug in more complex algorithms that may become available later.

Chapter 4

Conclusion

Models of the social relationship among humans can be used as a basis for the representations maintained by agents in referral systems. Referral systems can have significant impact in the corporate world. Many social-network applications are already being employed by companies for various purposes, but the social information has not yet been fully exploited. In this chapter we discuss the related work and potential extensions of our application.

4.1 Related Work

Some related works to capture social-network information are as follows

4.1.1 Semantic Analytics on Social Network

This is another application for modeling social networks. Using social-network information to detect Conflicts Of Interests (COI) between the committee that reviews scientific paper and the authors of the paper is discussed by [Aleman-Meza et al., 2006]. The sources used to extract the social information are the bibliographic literature. FOAF documents containing user profiles are also used as a source for extracting social information.

4.1.2 Analyzing the Social Network Data

Some techniques for capturing the social information are discussed in [Milgram, 1967]. But the captured information is not represented in a machine understandable format. Some of the interesting research areas discussed in the book are identifying an individual's opportunity structure, stability of individual's position in social network and understanding the social behavior of the whole population.

4.1.3 MARS

MARS [Yu and Singh, 1999] is a referral system used to identify experts in a particular field. It uses software agents to automate the search completion. Here each user's document repository is used as a source for determining his interest. A TF-IDF index is implemented on the collection to identify a request. Some of the data extraction techniques discussed in MARS could be applied to our system. MARS also uses the underlying social network to provide referrals. MARS can use our bootstrapping process along with its existing procedure to capture social network information.

4.1.4 Social Phishing

Here we discuss privacy concerns that arise when people access information that is available in the web. Phishing is a way of acquiring sensitive information from a victim in a fraudulent way. Usage of the social network sites by attackers to get information about a victim and winning his trust are described in [Nathaniel]. This is an application to show how modeling social network can be used in a bad way.

Our approach is different from exisitng systems like ReferralWeb [Kautz et al., 1997], MARS [Yu and Singh, 1999], and MINDS, in which only the online document repositories and emails are used for extracting social information. Also the technique used to extract social information is specific to the sources considered. In our system, social networking sites which are each a central repository of online profiles of users, are used for constructing the social information. Another goal of our project is to develop a common social-network information extraction technique to be useful across various sources. Our approach also aims to represent the information as a FOAF, which would be useful for bootstrapping semantic web.

In identifying the domain expertise of the user the systems like MARS make use of TF-IDF, but our application now uses a Webservice to extract the keyword. But using Webservices give us the oppurtunity to plug in a more sophisticated service at a later point of time.

4.2 Directions

We have identified the directions in which our work could be improved or extended.

4.2.1 Inferring the Structure of Information Sources

In our approach, a programmer specifies the XPath expressions to transform the information in each input source to a format satisfying our input schema. Developing an approach to automatically infer the structure information would remove the need to get XPath expressions. Building such an application is not trivial. However, the process can be facilitated if the system can understand the context of elements in different information sources.

4.2.2 Sophisticated Algorithms for Domain Extraction

Our prototype counts the occurrences of keywords in email messages to measure the expertise value for a user. Making use of a more complex algorithm like TF-IDF in our system, would improve the accuracy of the measurement. Our system does not let the user to specify his interests dire ctly. Therefore, providing an interface to obtain the interest information from the user would make the application have a wider reach.

Bibliography

Boanerges Aleman-Meza, Meenakshi Nagarajan, Cartic Ramakrishnan, Li Ding, Pranam Kolari, Amit P. Sheth, Ismailcem Budak Arpinar, Anupam Joshi, and Tim Finin. Semantic analytics on social networks: experiences in addressing the problem of conflict of interest detection. In *WWW*, pages 407–416, 2006.

A. Culotta, R. Bekkerman, and A. McCallum. Extracting social networks and contact information from email and the web, 2004.

Olivier Y. de Vel, A. Anderson, M. Corney, and George M. Mohay. Mining email content for author identification forensics. *SIGMOD Record*, 30(4):55–64, 2001.

FOAF. http://xmlns.com/foaf/0.1/.

Jabber. http://www.jabber.org/.

Henry Kautz, Bart Selman, and Mehul Shah. Referral web: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.

Bruce Krulwich. Learning user interests in heterogeneous information repositories. In *Next Generation Information Technologies and Systems*, pages 0–, 1995.

Stanley Milgram. The small world problem. *Psychology Today*, 1:61–67, May 1967.

Tom Jagatic Nathaniel. Social phishing. URL `citeseer.ist.psu.edu/763474.html`.

Kathleen M. O'Connor and Stephen Sauer. Recognizing Social Capital in Social Networks: Experimental Results. *SSRN eLibrary*, 1905.

RDFDS. http://www.xulplanet.com/downloads/rdfds/.

Young-Woo Seo and Byoung-Tak Zhang. Learning user's preferences by analyzing web-browsing behaviors. In *AGENTS '00: Proceedings of the fourth international conference on Autonomous agents*, pages 381–387, New York, NY, USA, 2000. ACM Press. ISBN 1-58113-230-1.

Wen tau Yih, Joshua Goodman, and Vitor R. Carvalho. Finding advertising keywords on web pages. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 213–222, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-323-9.

B. Yu and M. Singh. An multiagent referral system for expertise location, 1999.