ONTOLOGY ENGINEERING AND FEATURE CONSTRUCTION FOR PREDICTING FRIENDSHIP LINKS AND USERS' INTERESTS IN THE LIVE JOURNAL SOCIAL NETWORK

by

VIKAS BAHIRWANI

B.E., Rajiv Gandhi Proudyogiki Vishwavidyalaya, India, 2005

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2008

Approved by:

Co-Major Professor
William Hsu

Co-Major Professor
Doina Caragea

Copyright

Abstract

An ontology can be seen as an explicit description of the concepts and relationships that exist in a domain. In this thesis, we address the problem of building an interests' ontology and using the same to construct features for predicting both potential friendship relations between users in the social network *Live Journal*, and users' interests. Previous work has shown that the accuracy of predicting friendship links in this network is very low if simply interests common to two users are used as features and no network graph features are considered. Thus, our goal is to organize users' interests into an ontology (specifically, a concept hierarchy) and to use the semantics captured by this ontology to improve the performance of learning algorithms at the task of predicting if two users can be friends. To achieve this goal, we have designed and implemented a *hybrid clustering* algorithm, which combines hierarchical agglomerative and divisive clustering paradigms, and automatically builds the interests' ontology. We have explored the use of this ontology to construct interest-based features and shown that the resulting features improve the performance of various classifiers for predicting friendships in the *Live Journal* social network. We have also shown that using the interests' ontology, one can address the problem of predicting the interests of *Live Journal* users, a task that in absence of the ontology is not feasible otherwise as there is an overwhelming number of interests.

Table of Contents

List of Figures

List of Tables

Acknowledgments

Non doubtfully, my thesis represents the outcomes of my research at Kansas State University (KSU), but at the same time it reflects my interactions with generous and inspiring people that have contributed to my progress as a graduate student at KSU.

Dr. Doina Caragea, assistant professor with the department of *Computing and Information Sciences* at KSU and co-major adviser for my thesis, has taught me to think outside the box and strive to understand what is latent. The idea of developing an ontology of interests of users of the *Live Journal* social network is a result of many invaluable discussions I had with her. I wish to acknowledge her participation in the work presented in this thesis, and thank her for showing faith in me and guiding me in times of failure and success.

Dr. William H. Hsu, associate professor with the department of *Computing and Information Sciences* at KSU and co-major adviser for my thesis, has educated me with fundamentals of *Machine Learning*, which have been very useful in my research. I wish to thank him for his advises and for guiding me throughout my graduate program.

I am also thankful to Dr. Scott Deloach, associate professor with the department of *Computing and Information Sciences* at KSU and member of my thesis committee, for generously offering his time and expertise to better my work.

Furthermore, I am grateful to Dr. Virgil Wallentine, professor and head of *Computing and Information Sciences* at Kansas State University for his good natured support and for providing me with the resources needed to materialize ideas into an accomplished thesis.

At last, I wish to thank all my colleagues especially Tim Weninger, Waleed Aljandal, Swarnim Kulkarni and Abhishek Rakshit for their valuable discussions, comments and suggestions.

The research described in this thesis was supported by a grant from the National Science Foundation NSF 0711396.

Dedication

I dedicate this thesis in honor of my father, Dr. Ramesh Bahirwani, who has taught me that perseverance and perspiration are milestones to success.

I also dedicate it to my mother, Mrs. Vinita Bahirwani, who has taught me that great tasks are not accomplished by themselves, they rather begin with a single step.

My twin, Vishal Bahirwani, has always supported me in both professional and personal aspects of my life. This thesis would not be complete without dedicating it to him as an acknowledgment for his faith in me.

Chapter 1

Introduction

An ontology is an explicit, formal specification of a shared conceptualization of a domain of interest [Gruber, 1994], where "formal" implies that the ontology should be machine readable and "shared" implies that it is accepted by a group or community [Buitelaar et al., 2003]. In other words, an ontology is an explicit description (similar to the formal specification of a program) of the concepts and relationships that exist in a domain [Gruber, 1993]. Ontologies can be seen as high-level "umbrella" structures, which can be inherited and extended in order to organize data and knowledge [McGuinness, 2003]. They can also be seen as metadata that are used to provide a better understanding of the data or to facilitate data integration [Caragea et al., 2005; Doan and Halevy, 2005; Eckman, 2003; Levy, 2000; Noy and Stuckenschmidt, 2005].

Ontology-based data management systems (which organize data based on the semantic knowledge of a domain) find their applications in Web-based learning [Shridharan et al., 2004], software development (e.g., consistency checking, support and validation testing) [Happel and Seedorf, 2006; McGuinness, 2003], flexible querying of heterogeneous data sources [Calvanese et al., 2005], etc. In social networks, ontologies can provide a crisp semantic organization of the knowledge available in such networks. For example, users' interests can be grouped in a concept hierarchy that makes explicit the *implicit* relationships between various interest concepts, thus helping in the process of data understanding and analysis.

In previous work, Hsu et al. [2006] have addressed the task of predicting if two users

of the *Live Journal* social network can be friends or not. *Live Journal* is an online journal service with an emphasis on user interaction [Fitzpatrick, 1999]. Among other things in the *Live Journal* social network, users can specify their demographics, list their interests and tag other users as their friends. Given the emphasis on user interaction, *Live Journal* network can be represented by a graph structure, wherein nodes of the graph represent the users of the journal service and edges represent friendship links. Thus, if user "A" tags user "B" as a friend, then there exists an edge between corresponding nodes "A" and "B" in the underlying graph. The task of predicting friendship between two users is the task of predicting these friendship links.

Hsu et al. [2008] implemented a data acquisition tool known as "LJCrawler" to gather user data from the *Live Journal* website "www.livejournal.com". LJCrawler (version 3) is a multi-threaded, parallel HTTP crawler that effectively gathers user information in a breadth-first manner. Data acquisition began with *Live Journal* user "darthvader" who was randomly selected. The crawler queried user data including, but not limited to, user age, interests, friends, schools, communities, etc. and generated *raw LJ crawl data*. Furthermore, this raw data was post processed to delineate the *Live Journal* graph, and extract a set of graph-based and interest-based features [Hsu et al., 2006]. Hsu et al. [2008] have addressed the task of predicting friendship links using these features. Their results suggest that features constructed only from the common interests of two users are ineffective at predicting friend relationships. While network graph features prove to give good results, several questions can be still raised: Can we improve the performance of the algorithms that use graph features by combining them with interest-based features? If we are interested in other prediction problems, such as the prediction of the users' interests themselves, how can we handle the overwhelming number of interests? Furthermore please note that, a social network may be incomplete e.g. If two users "A" and "B" are friends in real life, but they have not tagged each other as friends in the social network, then links between nodes "A" and "B" will be missing from the graph. As a consequence, the graph features derived from the graph will be

incomplete or simply unavailable. In such cases, can the task of predicting friendship links be addressed? We hypothesize that organizing interests into an ontology and constructing interest-based features using the ontology can help to successfully address some of these questions.

To illustrate the importance of incorporating semantic information in the data used by prediction algorithms, we consider a simple example. Suppose that a user A is interested in "laptops" and a user B is interested in "desktops". A naive learner will consider "laptops" and "desktops" to be two different entities due to their lexical differences - a semantically incorrect assumption. If an interest ontology was available, "laptops" and "desktops" would probably be grouped together into a more abstract concept called "computers", and thus, the learner would be able to semantically link the more specific concepts "laptops" and "desktops" and use this information in the prediction process.

It has been previously shown [Hotho et al., 2003; Zhang et al., 2005] that the use of metadata (e.g., concept hierarchies) in addition to data, can improve the quality (accuracy and interpretability) of the learned predictive models in several domains. In this thesis, we investigate the use of ontology to improve the performance of learning algorithms at predicting friendship links in a social network. We also explore the effectiveness of ontology at the task of predicting interests of *Live Journal* users. To do this, we will construct an ontology based on declared interests of users in the social network *Live Journal*.

The effectiveness of this work has been already demonstrated in preliminary studies published in two conference papers. The first paper [Bahirwani et al., 2008], published in the Proceedings of the 2^{nd} ACM Social Network Mining and Analysis workshop (SNA-KDD 2008), has focused on organizing users' interests in an ontology and using the semantics captured by this ontology to improve the performance of learning algorithms at predicting if two users can be friends. We have empirically shown that the ontology so produced is very useful in predicting friendships in the absence of graph features. In addition, the combination of graph-based features and interest-based features derived in presence of the

ontology is the most effective in addressing this problem.

The second paper [Aljandal et al., 2008] to which this thesis work has contributed, was published in the Proceedings of the Artificial Neural Networks in Engineering conference (ANNIE 2008), and has explored the effectiveness of interestingness measures derived for the association rules based on common interests of users of the *Live Journal* social network, at predicting friendship links in the same. The results indicate that classifiers show an improvement in their prediction performance, when presented with association rule based measures refined in the presence of the interests' ontology, compared with their performance using the same set of features in absence of the ontology, thus suggesting the effectiveness of using the ontology at that task.

This thesis presents an extension of these preliminary studies and is organized as follows: Chapter 2 presents background information for the problem addressed. Chapter 3 introduces the details of the algorithm for ontology building. Chapter 4 describes the procedure used to evaluate the ontology extracted from the data. Chapters 5 and 6 discuss the experiments performed and the results obtained when predicting friend relationships and interests of users using the ontology, respectively. A discussion of the related work can be found in Chapter 7. We conclude and present several directions for future work in Chapters 8 and 9 respectively.

Chapter 2

Background

An ontology can be seen as an explicit description of the concepts and relationships that exist in a domain. The work presented in this thesis focuses on: (1) building ontologies (or a concept hierarchies) of interests specified by the users of *Live Journal* social network; and (2) using the derived ontologies to address prediction problems such as prediction of friendship links underlying the network graph and prediction of interests of users of *Live Journal*. Clustering algorithms help us find concepts in the data by providing mechanisms for grouping similar entities together. Moreover, clustering algorithms such as hierarchical clustering organize data entities in the form of hierarchy of concept-clusters providing the ability to explore the derived concepts at various levels of abstraction. We will present a general description on "Clustering Algorithms" in Section 2.1, followed by a description of the shortcomings of some approaches considered for engineering the interest ontology in Section 2.2.

2.1 Clustering Algorithms

Clustering is an unsupervised learning task that can be defined as the process of partitioning data into groups of similar entities, where each group corresponds to a concept [Berkhin, 2002]. Clustering algorithms are highly dependent on the selection of a distance metric that assigns a score to every pair of entities that may be grouped together. The distance metric captures the extent of similarity (or dissimilarity) between candidate pairs. The

distance between two clusters is usually computed as the average, maximum or minimum distance among all distances between the possible pairs of entities contained in the two clusters. Based on the same, clustering algorithms can be classified as *average-linkage* (or group-average) clustering, *single-linkage* (or single-link) clustering and *complete-linkage* (or complete-link) clustering algorithms respectively [Manning et al., 2008, Chapter 17].

In addition, clustering algorithms can also be classified based on partial or complete membership of data entities within their corresponding clusters. In *hard* clustering algorithms, each data instance is a full or complete member of one or more clusters. *Partitional* clustering is a special case of hard clustering where each data instance is a full member of only one cluster. *Soft* clustering algorithms, on the other hand, compute a distribution of membership of a data instance over all clusters [Manning et al., 2008, Chapter 16]. An example of soft assignment is that an instance "Chinese cars" may have a partial (or fractional) membership of 0.5 in each of the two clusters "China" and "Cars", and a membership of 0 in all other clusters. The same instance, in a hard assignment, will have a complete membership of 1 in one or both the clusters "China" and "Cars", and a membership of 0 in all other clusters. Furthermore, according to Manning et al. [2008, Chapter 16], clustering algorithms can be divided into *exhaustive* and *non-exhaustive* clustering algorithms depending upon the number of data instances assigned to one or more clusters. Exhaustive clustering assigns each data entity to a cluster while non-exhaustive clustering may leave some entities unclustered.

Also, depending upon the manner in which the clusters relate to each other; clustering algorithms can be divided into *flat* clustering and *hierarchical* clustering algorithms [Manning et al., 2008]. Flat clustering algorithms create flat sets of clusters without any explicit structure that would relate the clusters with each other. Their counterpart, *hierarchical clustering* algorithms build a tree of clusters by successively grouping the closest cluster pairs, until no further grouping is possible. In the resulting tree (often called *dendrogram*), each cluster node at an intermediate level is associated with a parent cluster, one or more

child nodes and one or more sibling nodes. The hierarchical clustering approach allows the exploration of the data at different levels of granularity. Thus, parent nodes represent abstract notions of the detailed concepts that their children embody [Berkhin, 2002; Manning et al., 2008].

Hierarchical clustering approaches are further categorized as *agglomerative* and *divisive* according to their algorithmic structure and operation [Jain et al., 1999]. The agglomerative approach starts by considering each instance as a distinct singleton cluster, and based on the similarity criterion, successively merges clusters together until the termination conditions are satisfied [Jain et al., 1999]. The divisive approach begins with a root cluster that represents all data instances, and successively splits the clusters based on the cohesiveness (or dissimilarity) of the instances constituting the cluster members, until the termination conditions are met [Jain et al., 1999].

Fig. 2.1 delineates both forms of hierarchical clustering algorithms. The divisive algorithm, as described above, begins by dividing the root cluster into two clusters "A" and "B". Cluster "A" further divides into clusters "C" and "D", however, since the cohesiveness measure of cluster "B" exceeds its threshold (for details, please refer to section 2.2), cluster "B" is not further divided. The agglomerative algorithm begins with singleton clusters "A", "B" and "C", and builds the hierarchy bottom-up as shown in fig. 2.1.

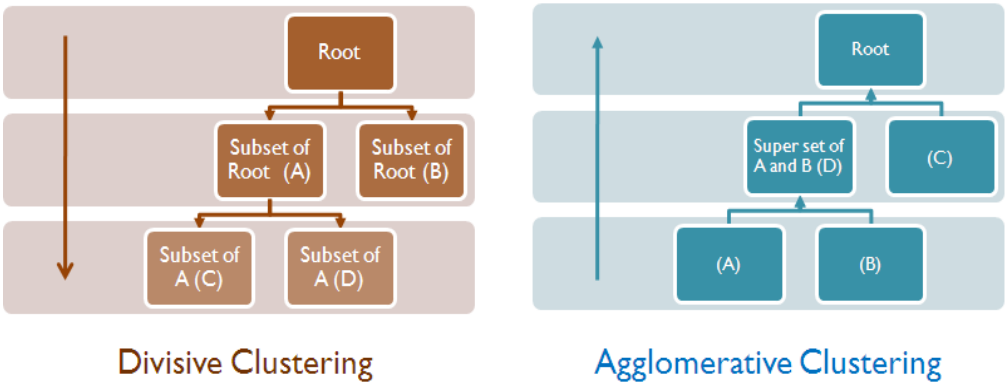

Figure 2.1: *Divisive(top-down) vs. Agglomerative(bottom-up) Hierarchical Clustering Paradigms*

We have designed and implemented a hybrid clustering algorithm, called *Hierarchical Agglomerative and Divisive* (HAD) clustering, to *automatically* organize the users' interests into a concept hierarchy. It can be classified into the hard, exhaustive and hierarchical clustering paradigms. Several other approaches were considered for this task. We will briefly describe them and their limitations in the next section and later show how our HAD algorithm overcomes these limitations in Chapter 3.

2.2 Clustering Algorithms Considered for Building an Ontology and their Shortcomings

There has been a lot of undergoing research in learning implicit concepts and concept hierarchies from the data using clustering approaches. Kang et al. [2004] introduce a hierarchical agglomerative clustering algorithm (Section 2.1) for construction of attribute value taxonomies from the data. An attribute value taxonomy (AVT) is a hierarchical grouping of attribute values that reflects assumed or actual similarities among values in a domain of interest [Kang et al., 2004, Section 1]. In their paper, Kang et al. describe a simple algorithm *AVT-Learner* for automated construction of AVT from the data. AVT-Learner recursively groups values of the attributes based on *Jensen-Shannon divergence* [Slonim and Tishby, 1999] between class distributions associated with attribute values to construct a taxonomy. Moreover, results [Kang et al., 2004] show that classifiers, e.g. Naive Bayes classifier, achieve comparable or higher classification accuracies than those obtained otherwise, when using AVTs generated by AVT-Learner,

As opposed to bottom-up approach of Kang et al., Punera et al. [2006] present a genetic top down algorithm (ATG) for a completely automated construction of a hierarchy of classes, given a set of labeled data points. ATG computes similarity between set of class labels using *Jensen-Shannon divergence* [Slonim and Tishby, 1999] and builds an n-ary hierarchy of classes following the divisive clustering paradigm (Section 2.1). Furthermore, Punera et al. [2006] not only quote that n-ary tree based taxonomies are more "natural" than binary tree

based taxonomies, but their results also show that n-ary taxonomy aware classifiers yield better classification accuracies as opposed to classifiers based on their binary counterparts.

In other related work, Kim and Chan [2003] address the problem of modeling users' interests by classifying the web pages that a particular user visits. They introduce a top-down divisive hierarchical clustering method (DHC) to recursively divide parent clusters into child clusters until a termination criterion is met. This division is based on the similarity of two interests in the parent cluster. The similarity of two interests is based on *content* (i.e., words and/or phrases describing interests). For each pair of interests, a similarity function [Kim and Chan, 2003, Section 4.2] is used to produce a score. If the score exceeds a threshold, all similar interests are separated out from the parent in a child cluster. The termination condition for the recursive partitioning is satisfied when no parent cluster can be further divided because of lack of similarity among the interests it contains.

Godoy and Amandi [2005] describe a practical way to implement the method introduced by Kim and Chan [2003]. Among the clustering approaches mentioned above and other similar approaches; the approach presented by Godoy and Amandi [2005], in particular, was very attractive to our work as they present an unsupervised web document clustering algorithm (WebDCC) that performs incremental concept learning, and organizes the concepts learned into a hierarchy. The procedure begins with a root cluster, which represents "everything". Initially, instances are added one-by-one to this root cluster and with each addition, the cluster is evaluated by an *evaluation function* [Godoy and Amandi, 2005, Section 2] for its *cohesiveness*. The evaluation function assigns a numerical score to each cluster, that represents the extent of similarity among the entities constituting that cluster. Lower value of the cohesiveness measure indicates that the cluster does not provide enough information to define a concept which summarizes its constituting instances, e.g. the cluster only consists of a few instances. Contrary to this, if the cohesiveness measure exceeds a given threshold, then the constituting instances are separated into a new child cluster. Moreover, Godoy and Amandi [2005, Section 2.4] empirically determine the value of this threshold to be 0.25.

In addition, every cluster in WebDCC algorithm [Godoy and Amandi, 2005], except for the root cluster, has a *concept description* associated with it. This concept description is a set of terms that describe the instances in the corresponding cluster. The approach proposed by Godoy and Amandi [2005] requires each instance "entering" into the clusters below the root to be compared with the associated clusters' concept descriptions. WebDCC algorithm assigns a numerical score to this comparison as described in Section 2.2 [Godoy and Amandi, 2005, Equation 2]. If the similarity measure between a new instance and a concept description exceeds a given threshold, then the instance is added to the cluster associated with that concept. However, if the similarity measure does not exceed the threshold, then the instance is either compared against other peer concept descriptions, or added to the parent cluster, if WebDCC fails to find any peer cluster to which the new instance can be assigned. The results presented in section 2.2 of the paper [Godoy and Amandi, 2005] suggest a value of 0.7 for this threshold. Thus, this idea of recursive partitioning extends from root to leaves, and partitioning of a cluster takes place when a newly added instance increases the cohesiveness of the cluster above a threshold [Godoy and Amandi, 2005].

Our initial approach to building an interest ontology was based on the notion of similarity introduced in [Kim and Chan, 2003] and the algorithm proposed in [Godoy and Amandi, 2005], but it presented several shortcomings in the social network domain. First, the two papers [Godoy and Amandi, 2005; Kim and Chan, 2003] consider a bag of words (BoW) approach to describe instances (web pages), and thus each instance is represented as a fixed length array. In the social network context, users' interests can be seen as instances, but cannot be naturally represented by a fixed length (bag of words) array, as every user has a relatively small number of interests compared to the total number of possible interests.

Second, the approach in [Godoy and Amandi, 2005] does not allow instances to belong to several concepts. An instance is restricted to belong to only one of the learned concepts. However, this restriction is not desirable for interest concepts, as interests like "notebooks" can belong to more than one category, for example "school supplies" and "computers".

Third, if a cluster's cohesiveness measure exceeds cohesiveness threshold to produce a concept, but the concept description cannot be formed (because, maybe, all valid terms have been used to describe its parents), then further expansion of the conceptual hierarchy from this cluster will not be possible. This is because the associated concept description will not have any terms, and hence the similarity measure between any new instance and this concept description will be 0 [Godoy and Amandi, 2005, Equation 2].

Finally, WebDCC algorithm [Godoy and Amandi, 2005] tries to derive a concept description out of a cluster every time a new instance is added to it. If a concept is generated, new instances will be added to a cluster below this concept only if they are similar to the concept description. The similarity between the concept description and the new instance is evaluated as a dot product between two vectors representing them [Godoy and Amandi, 2005, Equation 2]. With that said, if instances in a cluster contain long definitions (description of an interest) i.e. definitions involving a lot of terms, the associated concept description will contain a lot of terms and hence will favor instances with long definitions to pass through it (the same can happen if small concept descriptions are compared with instances having small definitions). This shortcoming can be addressed by considering other measures of similarity such as cosine similarity [Manning et al., 2008, Section 6.3], however, based on other considerations mentioned above, we decided not to use the approaches in [Kim and Chan, 2003] and [Godoy and Amandi, 2005] to construct an ontology over the interests of users in a social network.

Instead, we propose a hierarchical agglomerative and divisive clustering approach, which overcomes these issues and produces a useful ontology of interests. We evaluate this ontology with respect to the improvement in the performance of algorithms for predicting friend relationships among network users, when the algorithms are presented with features computed based on the ontology.

Chapter 3

Hierarchical Agglomerative and Divisive Clustering

As the name of the algorithm suggests, Hierarchical Agglomerative and Divisive (HAD) clustering algorithm is a hybrid between the hierarchical agglomerative and divisive clustering paradigms. The algorithm is designed to make the ontology extraction process as fast as possible and at the same time to produce a *sensible* and *useful* ontology. Primarily, it consists of three steps:

- In the first step, HAD fetches definitions of interests expressed by *Live Journal* users, from various sources such as WordNet-Online, IMDB and Amazon. Every definition of an interest forms an instance that will be included in the resulting ontology.
- The second step divides the instances into different clusters based on the sources from where the definitions are fetched and other factors such as "genres" of books or movies specified as interests.
- At the final step, HAD engineers the concept hierarchy in a bottom-up fashion to produce a tree whose root collectively represents all instances and whose nodes represent concepts at various levels of abstraction.

We describe the above steps in detail in Sections 3.1, 3.2 and 3.3 respectively.

3.1 Obtaining Interest Definitions

Social network data consisting of 1000 users, their interests and declared friends, is obtained from the *Live Journal*. Surprisingly, the 1000 users have nearly 22,000 unique interests, from which we derived approximately 45,000 unique individuals or instances, as explained below.

Each of the 22,000 unique interests is read from a text file and queried against different sources for potential definitions or descriptions. We seek information from WordNet-Online for the meanings of valid words, Internet Movie Database (IMDB) for definitions of movies, and Amazon.com for definitions of books via Amazon Associates Web Services (AWS). We have chosen to retrieve specific definitions for movies and books, because many user interests in our data are related to such concepts. Usually, an interest word can have more than one meaning, generating more than one instance. Furthermore, instances of the same interest word may belong to different parts of speech. At last, an interest may be a movie and/or a book with a specific genre associated with it. The definitions retrieved from the different sources capture such information associated with interests.

Figure 3.1 illustrates the process of obtaining interest definitions from the different sources mentioned above. For obtaining definitions, each interest is queried to *Wordnet-Online*, *IMDB* and *AWS* in no particular order. Definitions fetched from one or more of these sources form instances which will be organized by the next steps of HAD algorithm (see Sections 3.2 and 3.3) into a concept hierarchy.

As an example, we have three definitions for the interest "character".The first definition is obtained using WordNet-Online and results in the following instances for "character":

```
character,n,reference|character|formal|describing|qualifications|...
character,n,grapheme|graphic|symbol|written|symbol|used|represent|...
character,n,genetics|functional|determined|gene|group|genes...
character,v,engrave|inscribe|characters...
```

Two more definitions are obtained from IMDB and AWS, by querying these sources for

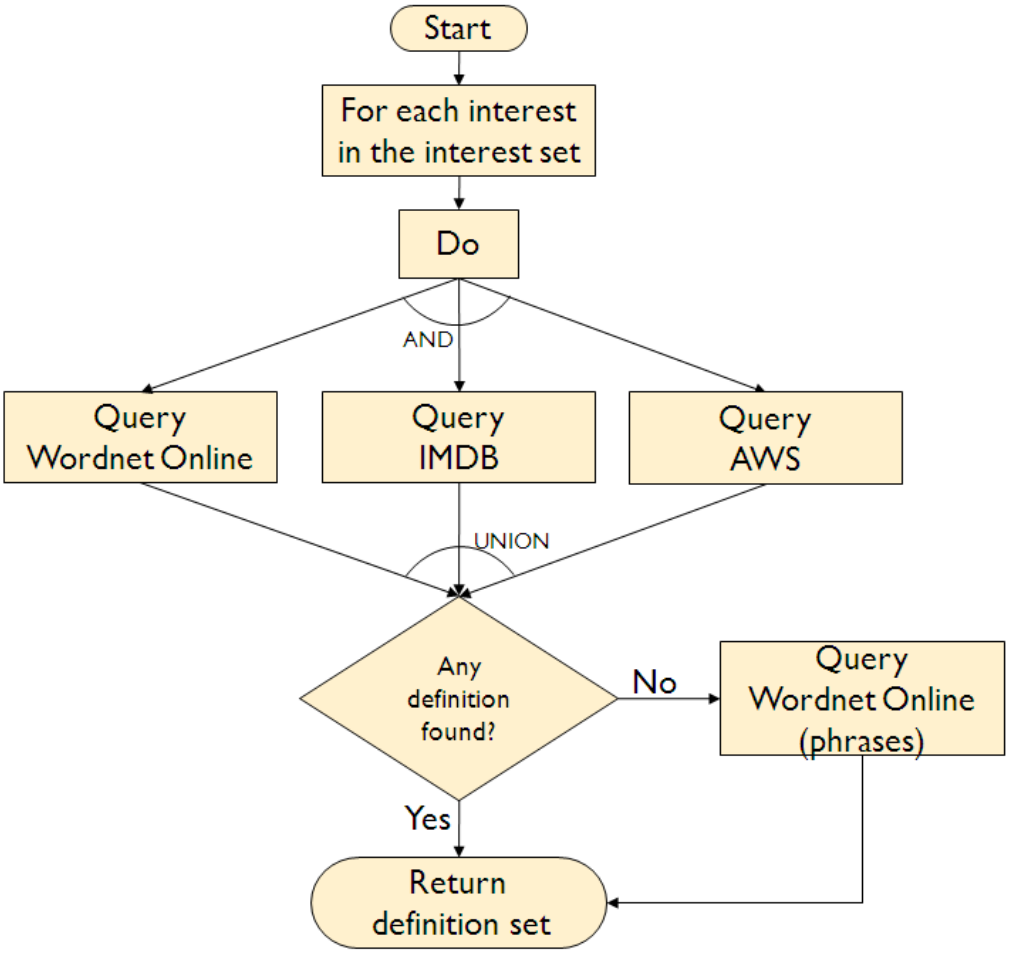

Figure 3.1: *Process of obtaining interest definitions*

"character" movies and books:

```
IMDB: character,reality tv,reality|film|fantasy|history|dream|rocky...
AWS: character,novel,butcher|covey|davenport|detective|dresden|files...
```

These definitions (also called as "individuals") are provided as input to the clustering algorithm and have the following format:

```
<interest>,<part of speech/genre>,<gloss>
```

where `<gloss>` is the set of words describing a particular interest. The gloss is extracted by filtering the text describing the interest, so that stop-words such as articles and prepositions are removed.

The following example shows the definitions of the interest "harry potter". Wortnet-Online does not provide any definition for "harry potter", however, IMDB and AWS do.

```
Wordnet Online: No definition found
IMDB:
harry potter,fantasy,adventure|chris|columbus|family|magic|hogwarts...
AWS:
harry potter,books,deathly|half-blood|harry|phoenix|potter|rowling...
```

As can be seen in Figure 3.1, for interests that are neither single words, nor movies/books (i.e. those interests for which none of the sources provide definitions), an "alternate definition" is formed. An alternate definition is the combination of definitions (fetched from WordNet-Online) of individual words that form the phrase for which the sources considered failed to provide definitions. For instance, the alternate definition for "aim pranks" is as follows:

```
aim pranks,alternate,aim|purpose|intention|design|pranks|buffoonery|
clowning|prank|acting|like|clown|buffoon
```

Apart from the interests that have valid definitions obtained from a variety of sources, users often specify interests that do not form valid interest words (e.g., "?????" or ":)"). There are approximately 500 such "interests" and we do not included them in the list of interests provided as input to the clustering algorithm. It is worth mentioning that for interests that have multiple definitions, HAD will consider these definitions as independent instances and will try to place them in relevant and possibly different clusters. This takes care of the major shortcoming described in Section 2, mainly that an interest can only belong to one cluster.

3.2 HAD: Divisive Clustering Step

After definitions for interests are fetched, the next step is to divide the resulting instances into four major clusters as shown in Figure 3.2. The first cluster consists of all the instances that are described in terms of meaningful words from WordNet-Online. The second cluster consists of movie definitions fetched from IMDB. The third cluster comprises of book definitions and the fourth contains instances with alternate definitions. About 22,000 unique interests queried for definitions generate 17,753 valid word definitions, 4,189 movie definitions, 18,168 book definitions and 1,986 alternate word definitions resulting in a total of 42,096 individuals to be clustered. Given the large number of book instances and the prior knowledge about genres, the "book" cluster is further divided into a set of sixteen sub-clusters based on genres (Action, Fantasy, Drama, Children, etc). Similar to books, movies can also be divided based on their genres. However, since there are only about 4,200 movie interests, compared to almost 18,000 book-interests, the "movie" cluster is not further divided in this phase of the algorithm; the grouping based on genre will be performed by the hierarchical agglomerative clustering.

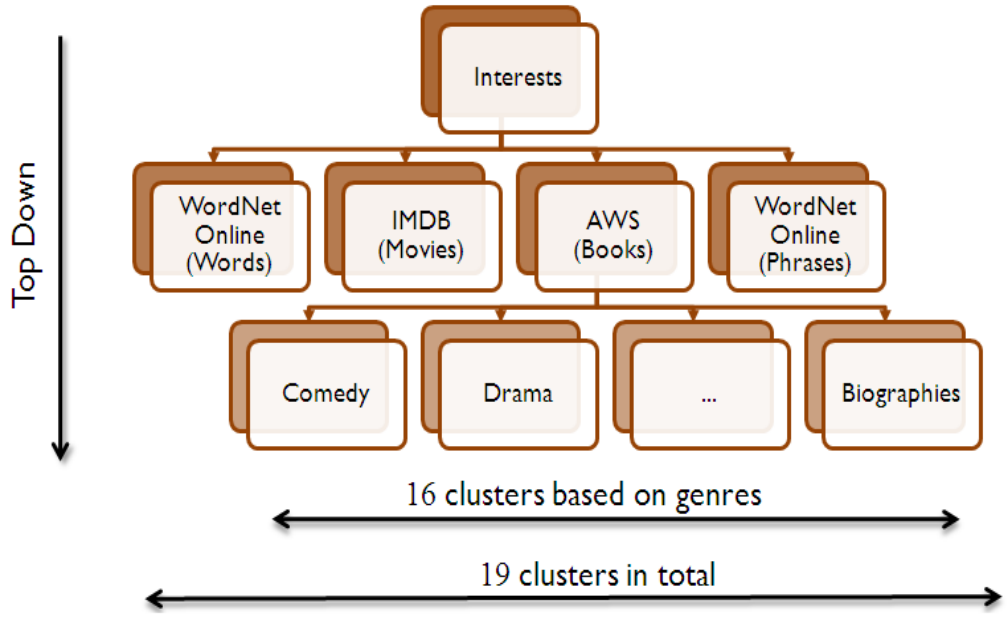

Figure 3.2: *HAD: Divisive Clustering step*

There are two advantages we gain by dividing the data into several clusters before applying the hierarchical agglomerative clustering algorithm. First, we can apply the algorithm in parallel to these clusters, resulting in faster ontology construction. Second, the source from which the definitions of an interest are obtained can inform us about other concepts that this interest can be associated with. The prior cluster division makes it possible to exploit this information.

3.3 HAD: Agglomerative Clustering Step

The hierarchical agglomerative clustering algorithm is independently applied to each of the nineteen clusters obtained in the divisive phase of HAD. It works in a bottom-up fashion and groups similar instances together in two ways as described in Sections 3.3.1 and 3.3.2, to engineer two versions of the interests' ontology.

Similarity between two instances (or equivalently, two singleton clusters) is defined as the number of common terms describing the instances. We evaluate this similarity as the dot product between vectors representing the instances. A binary representation where the i^{th} element in a vector is either set to 1 if the i^{th} term of the vocabulary is present in the definition of that instance, or is set to 0 otherwise, is used for each instance. As an example, Figure 3.3 shows the similarities among interests "Wireless", "Radio", "Wi-Fi" and "Networks". Furthermore, the similarity between two non-singleton clusters is considered to be the average similarity among pairs formed with elements from the two clusters (average linkage). A cluster is "matched" with another cluster if the similarity between the two clusters is maximum among all the possible pairs of clusters in the current set.

Following Sections 3.3.1 and 3.3.2 discuss two flavors of the hierarchical agglomerative phase of the HAD algorithm.

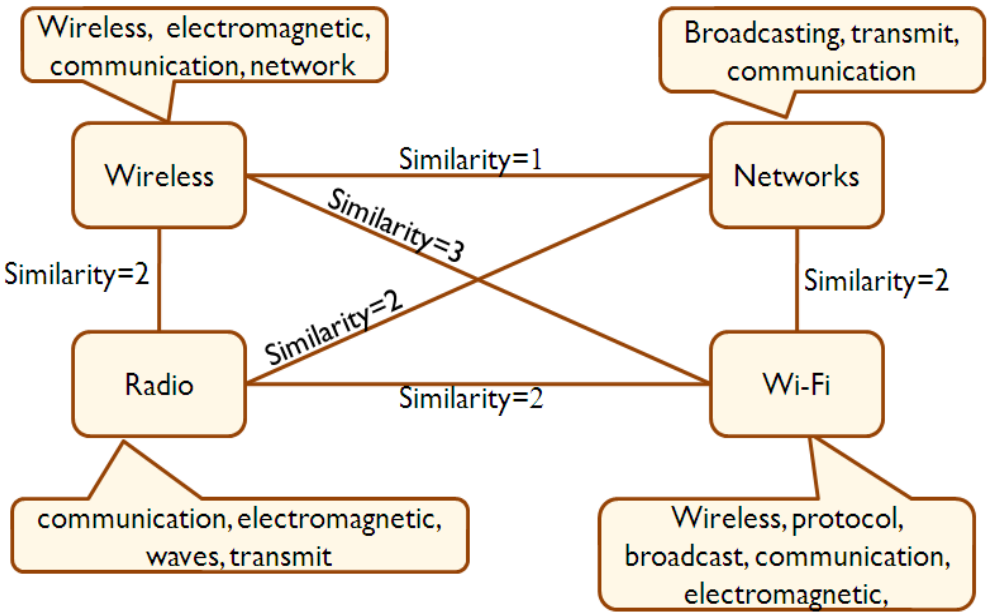

Figure 3.3: *Computing similarity between interests*

3.3.1 Agglomerative Step: version 1

This version of the agglomerative step takes as input the set of interest definitions (or instances) in a particular cluster. Initially, each instance is considered to be a singleton cluster and the set of all singleton clusters is called the "current" set. Agglomerative step (version 1) is said to complete an iteration when it has processed all the clusters constituting the "current" set as follows:

At each iteration, HAD considers every cluster present in the "current" set and aims to find another cluster matching it. If a match is found between two clusters, the two clusters are merged to form a new parent cluster, which will be added to the new "current" set to be used in the next iteration of the algorithm. However, if a cluster does not match with any other cluster i.e the similarity measure between this cluster and any other cluster in the "current set" is 0, then the cluster is added to the new "current" set as it is. Figure 3.4 shows an example of the agglomerative step described in this section. Figure 3.3 depicts that "Wireless" and "Wi-Fi" are have the maximum score of similarity metric among all possible scores between "Wireless" and other interests considered. Thus, "Wireless" and "Wi-Fi"

are grouped together. Similarly, "Networks" and "Radio" have a similarity measure of 2 (common terms being transmit and communication) and hence, are combined together. The agglomerative step continues to work in a bottom-up fashion and is said to convergence (or complete its job) in any of the following two cases: either the clusters in the current set do not match to any of their peers or the new current set has only one element. The version of the interests' ontology derived as a result of this flavor of the agglomerative step is denoted as "O1" in the rest of the thesis.

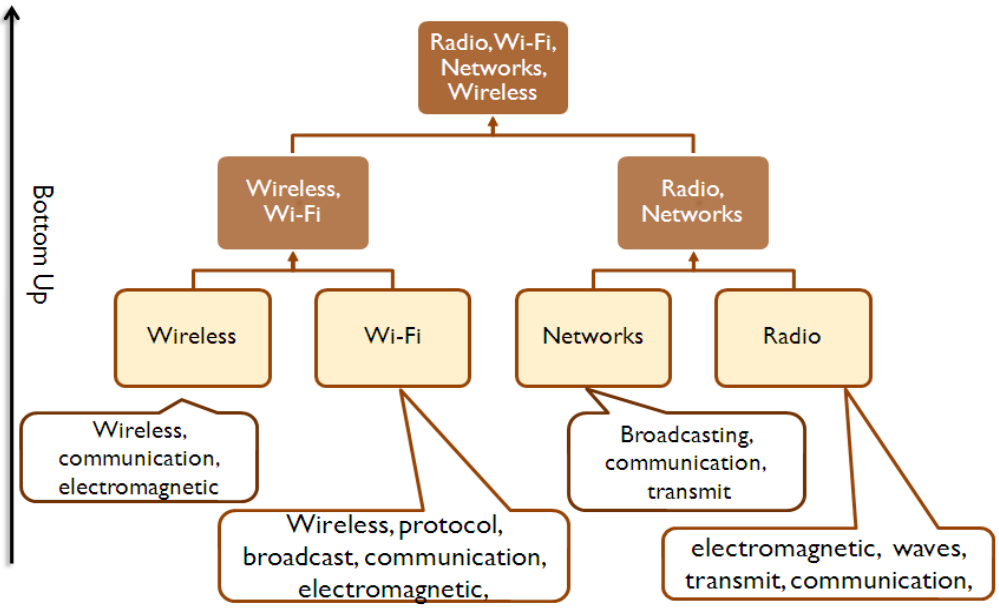

Figure 3.4: *Agglomerative Clustering step: Version 1*

3.3.2 Agglomerative Step: version 2

Similar to its counterpart (Section 3.3.1), this version of the agglomerative step takes as input the set of interest definitions (or instances) in a particular cluster. Initially, each instance is considered to be a singleton cluster and the set of all singleton clusters is called the "current" set. Agglomerative step (version 2) is said to complete an iteration when it has processed all the clusters constituting the "current" set as follows:

At each iteration, HAD considers every cluster present in the "current" set and aims to

find another cluster matching it. If a match is found between two clusters, the two clusters are merged to form a new parent cluster. Contrary to version 1 of agglomerative clustering step (Section 3.3.1), the new parent is added to the "current" set and is considered for further grouping in the *current iteration* of agglomerative clustering. However, if a cluster does not match with any other cluster i.e the similarity measure between this cluster and any other cluster in the "current set" is 0, then the cluster is added to the new "current" set as it is. The rest of the agglomerative step executes in a similar fashion as *agglomerative step - version 1.* In the following Chapters, we denote the version of the interests' ontology derived as a result of this flavor of the agglomerative step "O2".

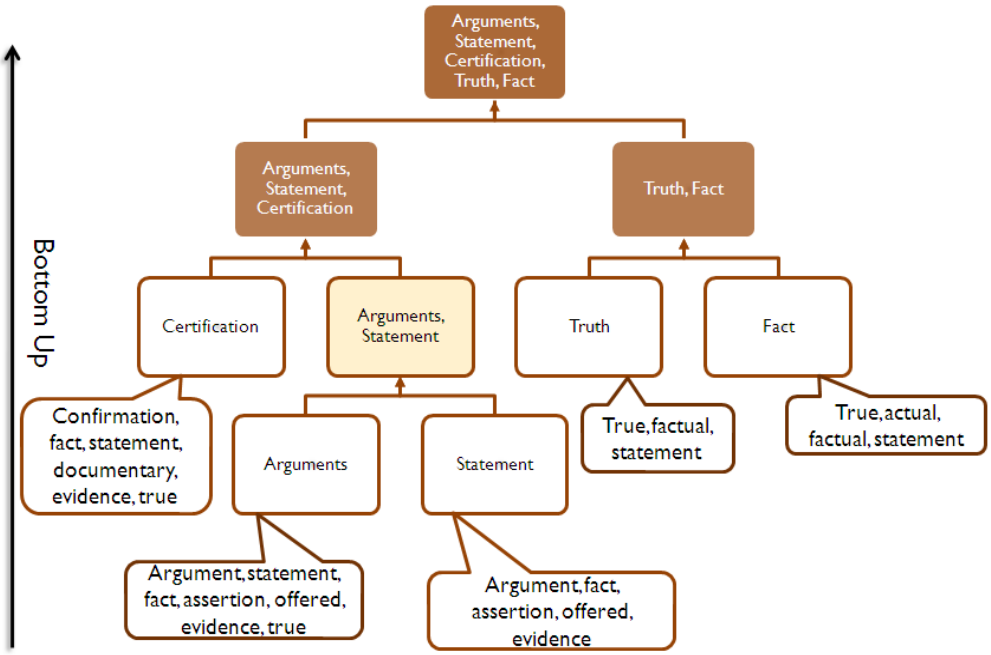

Figure 3.5: *Agglomerative Clustering step: Version 2*

Figure 3.5 shows the grouping of four interests "Truth","Certification", Arguments" and "Statement". The similarity measure between the latter two is maximum i.e. 6 (common terms being argument, fact, assertion, offered, evidence, true) among all possible scores between "Arguments" and other interests. Thus, "Arguments" and "Statement" are grouped together. Furthermore, as soon as the resulting cluster is formed, it is added to the "current"

set and later gets grouped with the interest "Certification" because the similarity measure between "Certification" and the newly formed cluster is maximum i.e. 4 among all possible scores between "Certification" other clusters. The agglomerative step continues until the algorithm converges to generate the ontology as shown in Figure 3.5.

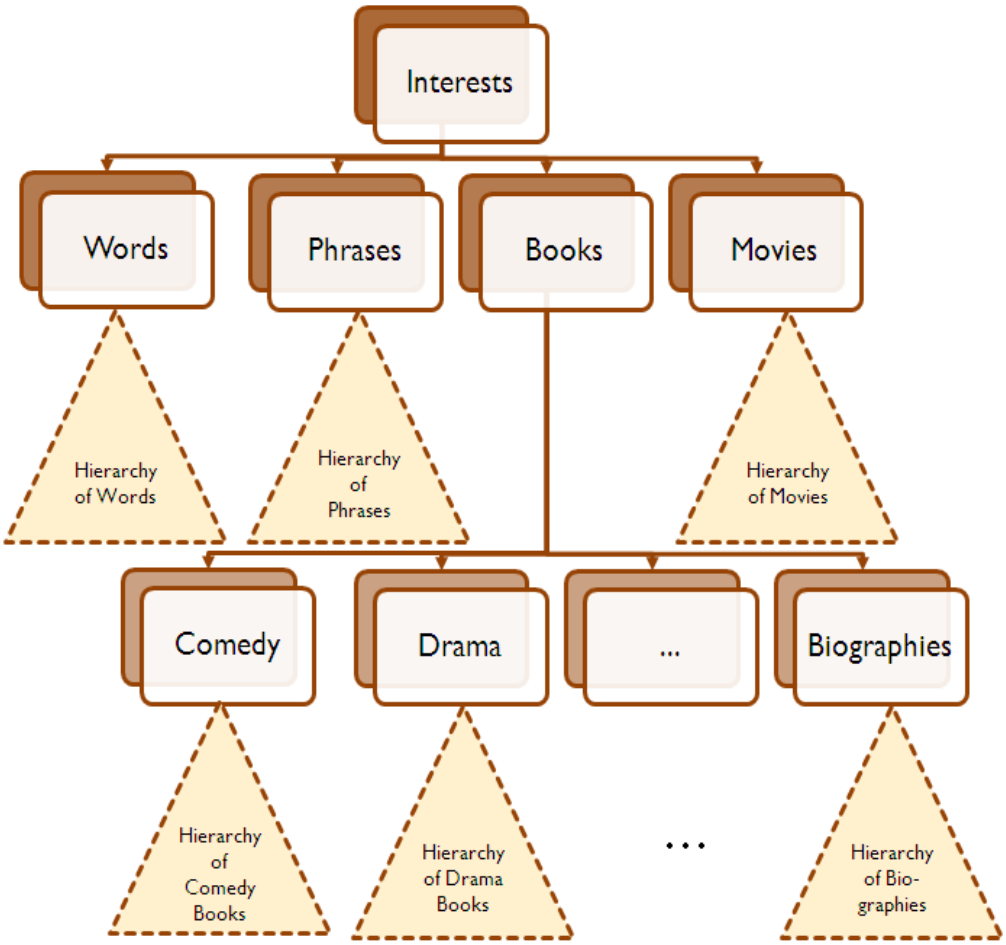

Figure 3.6: *Ontology of Interests*

After termination of either versions of the agglomerative clustering phase of HAD, all of the remaining clusters in the "current" set are combined to form one cluster which represents the root of hierarchy of input instances and their groupings. Like mentioned above, we have applied this algorithm to all nineteen clusters (obtained in the divisive phase of HAD), using a multi-threaded execution paradigm. Nineteen hierarchies obtained from all the

threads represent nineteen sub-ontologies of interests, one for each cluster resulting from the divisive phase (Figure 3.2). Each of these nineteen sub-ontologies is replaced with their corresponding clusters to generate the unified ontology of interests of users of *Live Journal* social network as shown in Figure 3.6.

Chapter 4

Evaluation of the Ontology through Learning

A prediction problem (often referred to as supervised learning) is defined as the task of inferring a function from a set of training examples [Mitchell, 1997]. The function to be learned is called the target concept (denoted by c), and the set of items over which it is defined is called the set of instances (denoted by X), i.e. $c : X \rightarrow \{class_1, class_2, class_3 \ldots class_k\}$, where $class_1$ to $class_k$ are k class labels associated with the set of instances X. For any instance $x \, \epsilon \, X$, the value $c(x)$ is called the class label for x [Mitchell, 1997]. In our work, $c(x)$ is a boolean valued function of the form $c : X \rightarrow \{0, 1\}$. When learning a target concept, the learner (or the classifier) is presented with a set of *training examples*, each consisting of an instance x from X, along with its class value $c(x)$. Instances for which $c(x) = 1$ are called *positive examples*, or the members of the target function. Instances for which $c(x) = 0$ are called *negative examples* or the nonmembers of the target concept. Given a set of training examples, the task of the learner is to estimate (or learn) c so that the learner can predict the class of a new instance (called the test instance) [Mitchell, 1997].

In addition, the performance of a learner is measured through the means of some performance measures such as *Precision*, *Recall*, *Accuracy*, *F-Measure* etc [Mitchell, 1997]. In our case, the performance of a learner is measured by the area under the *Receiver Operating Characteristic* (ROC) curve [Fawcett, 2005], i.e. the curve depicting the tradeoff between

the *true positive rate* vs. *false positive rate* (see Figure 4.1). The area under the ROC curve, or AUC (shaded region in Figure 4.1), is reported on as scale from 0 to 1, 0 being the minimum value and 1 being the maximum value. Thus, higher values of the AUC indicate better performances of a classifier at a given prediction task, while lower values indicate otherwise.

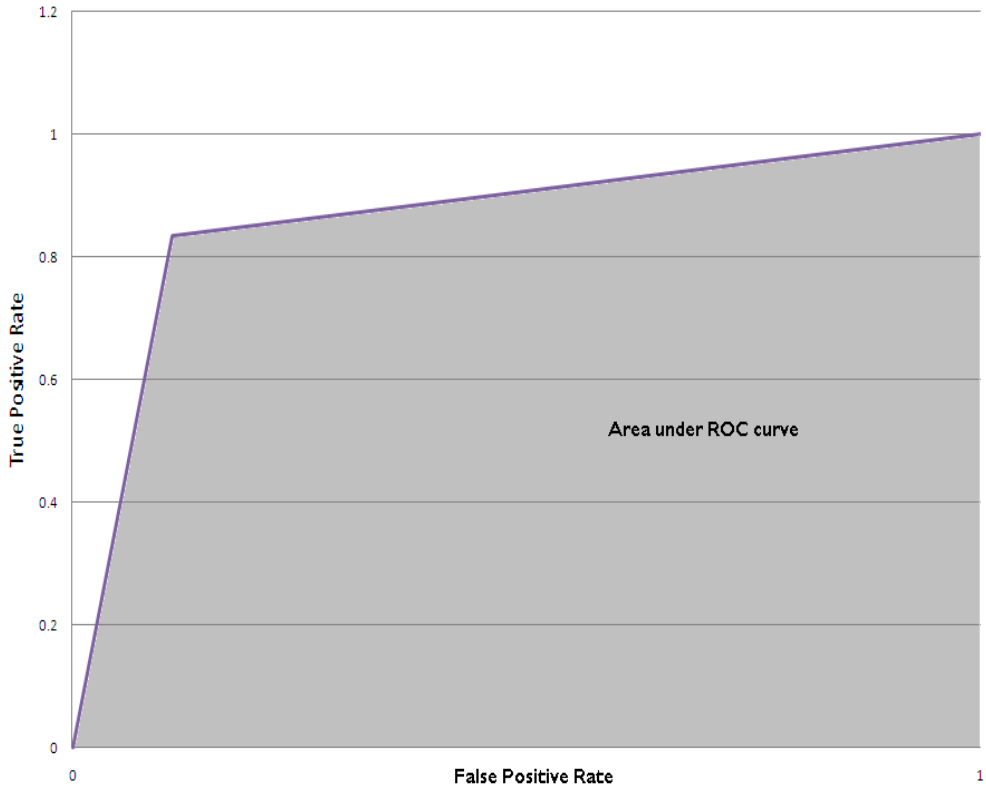

Figure 4.1: *Reverse Operating Characteristic (ROC) Curve*

The main goal of our work is to improve the performance of learning algorithms at the tasks of predicting friend relationships and interests of users in the *Live Journal* social network. We define the two prediction problems addressed formally as follows:

1. Predicting friendship relationships among *Live Journal* users.

    - Prediction task t: For a given pair of users, the task of a learner is to predict if the two users are friends or not ($c(x) \epsilon \{0, 1\}$).

- Set of instances X: The set of instances consists of examples, each of which is defined as $< user_A, user_B >$ where $user_A$ and $user_B$ identify the two users. Each instance $< user_A, user_B >$ can be represented as a vector of the form

$$< i_1, i_2, ..., i_k, g_1, g_2, ..., g_l >$$

where i_1 to i_k are values for k interest-based features that capture the information of common interests between $user_1$ and $user_2$, and g_1 to g_l are values for l graph-based features with respect to the user-pair for which the instance is defined (see Section 4.1).

- Class-label $c(x)$: For each instance x, $c(x) = 1$ if $user_1$ and $user_2$ are friends, or $c(x) = 0$ otherwise.

- Performance measure P: We use are under the ROC curve to measure the performance of learning algorithm at this task

2. Predicting interests of *Live Journal* users.

- Prediction task t: A user can be interested in more than one interest. Given a set of k interests, we learn k classifiers, one for each interest. The task of classifier$_i$ $(1 \leq i \leq k)$ is to predict if a given user is interested in interest$_i$.

- Set of instances X: The set of instances consists of examples defined as $< user_A >$ where $user_A$ identifies the user whose interests are to be predicted. Each instance $< user_A >$ can be represented as a vector of the form

$$< nom_1, nom_2, ..., nom_k, num_1, num_2, ..., num_l >$$

where nom_1 to nom_k are values for k interest-based nominal features and num_1 to num_l are values for l interest-based numerical features. Both nominal and numerical features summarize the information of common interests among the friends of $user_A$ for which the instance is defined (see Section 4.1).

- Class-label $c(x)$: Given k interests, for each classifier$_i$ $(1 \leq i \leq k)$, $c(x) = 1$ if $user_A$ is interested in interest$_i$, or $c(x) = 0$ otherwise.
- Performance measure P: We use the area under the ROC curve [Fawcett, 2005] (see Figure 4.1) and the hamming distances (see Section 5.2 Equation 5.1) to measure the performance of the learning algorithms at the task of predicting interests.

We will formally evaluate the ontology through the results of prediction algorithms that use interest-based features, constructed using the interest ontologies, alone or in combination with graph-based features derived from the network graph underlying *Live Journal*. In what follows, we provide a more detailed description of features that a briefly introduced above.

4.1 Types of Features

This section presents several types of features that can be derived from *Live Journal* data and can be used to address the prediction problems at hand.

1. *Interest-based features*:

   (a) Nominal features:

   - Predicting friendships: User interests themselves can be indicative of user friendships and could be used as *interest-based nominal features*. Intuitively, if two users share a "rare" interest, then the two users can potentially be friends. For the task of predicting friendships between a pair of users "A" and "B", interest-based nominal features consist of the interests of user "A" followed by the interests of user "B".
   - Predicting interests: *Interest-based nominal features* corresponding to the friends of a specific user can be used to predict interests of that user. For instance, the fact that a significant number of friends of user "A" are interested in "outdoor sports" can be indicative of the fact that user "A" is

also interested in "outdoor sports". Given a set of k interests, we address the problem of predicting interests by considering k nominal features (one for each interest). The value of the i^{th} nominal feature (where $1 \leq i \leq k$) is equal to the number of friends of user "A" that are interested in "interest$_i$".

Given the large number ($\approx 22,000$) of interests in our data, an instance consists of overwhelming number of interest-based nominal features. With that said, the classifiers may not perform well at predicting friendships or interests of users. However, one can try to use user interests at higher levels of abstraction in the interests' ontology and expect an improvement in the performance of the classifiers compared to their performance in the former case.

(b) Numerical features:

If two users have many interests in common, then it is possible that they are friends, regardless of what exactly those interests are. Several *interest-based numerical features* capturing this intuition can be derived and used to address the prediction tasks at hand. Section 4.2 presents an overview of our related work [Aljandal et al., 2008], where a variety of interest-based numerical features are investigated.

- Predicting friendships: Given a pair of users "A" and "B", the interests of the two users and hence mutual interests between the pair, we address the problem of predicting friendships by computing the numerical features as described in 4.2.
- Predicting interests: Given a user "A" and a set "F_A" of friends of "A", we compute (see Section 4.2) interest-based numerical features for every pair $\{user_A, user_x\}$ where $user_x \, \epsilon \, F_A$. Each of the numerical features computed is then averaged over the set "F_A" and used for predicting interests of user "A".

2. *Graph-based features*:

- Predicting friendships: The social network *Live Journal* is essentially a graph with nodes representing the users and edges representing the friendships among users. Similar to Hsu et al. [2006], we consider several graph-based features, listed below, for predicting friendship links in the *Live Journal* social network. For more detailed definitions of these features, please see [Hsu et al., 2006].

  To predict whether user "B" is a friend of user "A" i.e. there is an edge A→B in the social network graph, we consider the following graph-based features derived from the graph:

  (a) *In-degree* of "A" (InDeg(A)): The number of edges terminating at the node corresponding to user "A" represent the popularity of user "A" and are denoted as the *in-degree* of that user.

  (b) *In-degree* of "B" (InDeg(B)): Similar to *in-degree* of user "A", this is the number of incoming edges to the node corresponding to user "B" in the social network graph.

  (c) *Out-degree* of "A" (OutDeg(A)): The number of friends of user "A" except user "B" are denoted as the *out-degree* of user "A". These are computed by counting the number of outgoing edges (except the edge A→B) from the node corresponding to user "A" in the social network graph.

  (d) *Out-degree* of "B" (OutDeg(B)): Similar to *out-degree* of user "A", this represents the number of existing friends of user "B" except user "A".

  (e) *Forward deleted distance* between "A" and "B" (FDD(A,B)): The minimum alternative distance (i.e. not considering edge A→B if it exists) from the node corresponding to user "A" to the node corresponding to user "B" in the graph is denoted as the *forward deleted distance* between the two users.

  (f) *Backward distance* from "B" to "A" (BD(A,B)): The minimum distance from the node corresponding to user "B" to the node corresponding to user "A" in the graph is denoted as the *backward distance* between the two users.

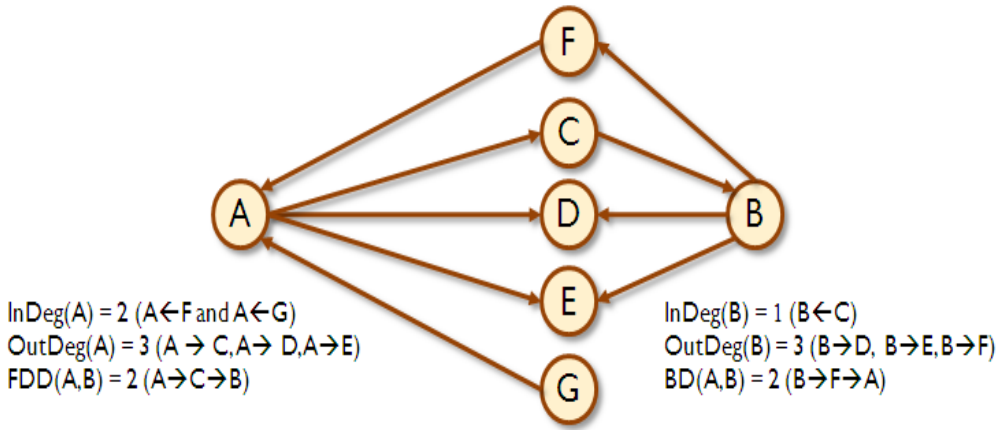

Figure 4.2: *Graph-based features*

Figure 4.2 shows a sample social network graph consisting of seven users (A to G) and the values for various graph-based features for the candidate user pair $\{A, B\}$.

- Predicting interests: Pair-wise graph features proposed by Hsu et al. [2006] cannot be directly applicable to predicting interests of users of *Live Journal*. Thus, we incorporate the link information retrieved from *Live Journal's* social network graph by considering the friends of each user, when computing interest-based features to address the task of prediction of interests of that user as described above.

To evaluate the usefulness of the interests' ontology in predicting friendships in *Live Journal* we compare the performance of several learning algorithms at this task, when used with interest-based features only, graph-based features only and a combination of the two sets of features, both in the presence and absence of the ontologies. In addition, to evaluate the usefulness of the interests' ontology in predicting interests of users of the *Live Journal* social network, we compare the performance of several learning algorithms at this task, when used with interest-based nominal features only, interest-based numerical features only and a combination of the two, both in the presence and absence of the ontologies.

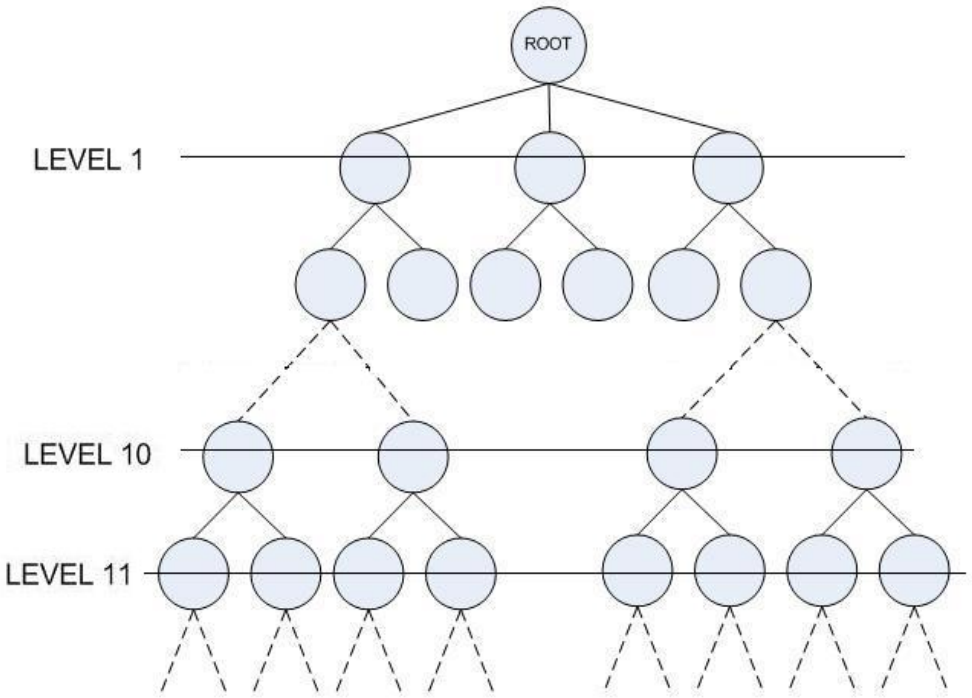

Figure 4.3: *Different levels of abstraction in an ontology*

When deriving interest-based features using the ontologies, interests of users are viewed at different levels of abstraction (Figure 4.3) and experiments are conducted to reveal the levels in the hierarchies that give the best performance (see Section 5).

4.2 Computing numerical interest based measures using Association Rules

Association rules are rules of the form A→B, where both A and B are subsets of an observed item set L = $\{I_1, I_2, \ldots, I_k\}$. In the field of association rule mining, a variety of measures for estimating rule interestingness have been proposed. Geng and Hamilton [2006] review 38 probability-based objective interestingness measures for association rules and summaries, classify them from several perspectives and compare their properties. In another survey, Tan et al. [2002] discuss the properties of 21 objective interestingness measures and conclude

that there is no "single" measure that is consistently better than others in all application domains. Each measure captures different rule characteristics in a domain.

In the context of *Live Journal*, each interest has an item set whose elements are users having that interest, and from each item set association rules of the form A→B (where A and B are users) can be formed. Aljandal et al. [2008] look globally in all the item sets and define eight objective measures of rule interestingness (listed below), from which eight interest-based numerical features are constructed and used alone or in combination with graph features to predict friends.

Suppose that $numInt_A$ denote the number of interests of user "A", $numInt_B$ denote the number of interests of user "B" and $numInt_{AB}$ denote their mutual interests. For the user pair $\{A, B\}$, we define the following probabilities:

- Probability of user "A" having an interest i.e.

$$P(A) = \frac{numInt_A}{totalnumberofinterests} \tag{4.1}$$

- Probability of user "B" having an interest i.e.

$$P(B) = \frac{numInt_B}{totalnumberofinterests} \tag{4.2}$$

- Probability of users "A" and "B" having a common interest i.e.

$$P(AB) = \frac{numInt_{AB}}{totalnumberofinterests} \tag{4.3}$$

Based on Equations 4.1, 4.2, 4.3 and the *Bayes' Theorem*, the following eight objective measures of rule interestingness can be derived:

1. $Support(A \rightarrow B) = P(AB)$
2. $Confidence(A \rightarrow B) = P(B|A)$
3. $Confidence(B \rightarrow A) = P(A|B)$

4. $Lift(A \rightarrow B) = \frac{P(B|A)}{P(B)}$

5. $Conviction(A \rightarrow B) = \frac{P(A)-P(\rightarrow B)}{P(A \rightarrow B)}$

6. $Match(A \rightarrow B) = \frac{P(AB)-P(A)*P(B)}{P(A)*(1-P(A))}$

7. $Accuracy(A \rightarrow B) = P(AB) + P(\rightarrow A \rightarrow B)$

8. $Leverage(A \rightarrow B) = P(B|A) - P(A)P(B)$

However, Aljandal et al. [2008] point out that one limitation of existing measures of rule interestingness is that they do not account for the relative size of item sets to which each candidate pair of associated subsets (A, B) belongs. In fact, there may be some hidden associations related to candidates appearing in small groups. "Giving some attention and weight to these small groups may lead us to a different relationship perspective" [Aljandal et al., 2008]. Some interests such as DNA replication (an example of "rare" interest) having low membership (small item set size) often suggest a more significant association between users listing them than between those who have interests such as Music or Games in common. In their paper, Aljandal et al. derive a normalization factor that takes into account the *popularity* of particular interests two users hold in common, with the most popular interests (held by a significant proportion of users) being slightly less revealing than rarer interests. For any given interest (say interest$_i$), the normalization factor R_i is computed according to the Equation 4.4:

$$R_i = \sqrt[q]{\log_{n_i} m} \tag{4.4}$$

where q and m are constants and n_i is the number of users interested in interest$_i$ (size of the item-set corresponding to that interest). Aljandal et al. [2008] empirically suggest the values of q and m to be 2 and 5 respectively. In addition, they show that the normalization factor can be incorporated into deriving eight measures of interestingness (listed above) by computing the number of interests of a particular user and common interests of a pair of users as follows:

- Number of interests of user "A" i.e.

$$numInt_A = \sum_{i=1}^{k} j \tag{4.5}$$

where k=number of interests, and $j = R_i$ if "A" is interested in interest$_i$, or $j = 0$ otherwise.

- Number of interests of mutual interests between users "A" and "B" i.e.

$$numInt_{AB} = \sum_{i=1}^{k} j \tag{4.6}$$

where k=number of interests, and $j = R_i$ if "A" and "B" both are interested in interest$_i$, or $j = 0$ otherwise.

The values of $numInt_A$ (similarly $numInt_B$) and $numInt_B$ so derived are then used to compute the probabilities from Equations 4.1, 4.2 and 4.3, and hence eight measures of interestingness. Aljandal et al. [2008] empirically show that when presented with normalized interestingness measures (as numerical features) for predicting friendship links in the *Live Journal* social network, classifying algorithms show an improvement in performance with respect to their performance at the same task when presented with unnormalized interestingness measures. Hence, we use normalized versions of the eight association rule measures mentioned above, along with the number of common interests, as features for the prediction problems at hand.

Chapter 5

Experimental Setup

This Chapter describes the details of our experiments conducted to evaluate the ontology of interests as discussed in Chapter 4. We have conducted two sets of experiments designed to investigate the performance of several classification algorithms at (a) predicting friend relationships and (b) predicting interests of users of *Live Journal*, when presented with different sets of features, including ontology based features.

In each experiment, we consider the following classifiers, whose implementations are provided by the WEKA data mining software [Witten et al., 1999].

- J48 decision tree:

  A decision tree classifier is represented in the form of a tree wherein non-leaf nodes represent conditional tests performed on one or more attributes of the data instances and leaf nodes represent the classification outcomes of the same. A decision tree is learned by splitting the training data set into subsets based on an a conditional test performed on an attribute value at each non-leaf node. This process is repeated recursively on each derived subset, and training process is said to complete either when further splitting of training data is not possible or all the examples in the derived subset are associated with the same class label. J48 is a simple implementation of a decision tree classifier [Mitchell, 1997].

- Support Vector Machines (SVM) with *build logistic model* option enabled:

SVM views input instances as two sets of n-dimensional vectors. The first set consists of positive examples while the second set comprises of the negative examples. Based on the training examples presented to the classifier, it tries to construct an n-dimensional hyperplane that separates the two sets. If more than one separating hyperplane can be constructed, then SVM considers the hyperplane which maximizes the margin between the two data sets [Mitchell, 1997].

- Random Forests:

  A random forest classifier uses a collection of decision trees to classify data instances. The predicted class label for each instance is same as the value of the class label predicted by majority of the decision trees in the random forest collection [Mitchell, 1997].

- Logistic regression (Logistic):

  Logistic regression classifier predicts the class of each data instance by computing the probabilities assigning that instance with a positive and negative class labels. These probabilities are computed by fitting the data instance to a logistic curve that is derived in the training phase of the learning process [Mitchell, 1997].

- One-attribute-rule (OneR):

  One attribute rule algorithm classifies data instances according to very simple association rules involving only one attribute in the *condition* part. OneR classifier aims to find one attribute that can be used to predict the class labels of new instances by making as few prediction errors as possible [Mitchell, 1997].

The rest of the Chapter presents the experimental design for the task of predicting friendship links in Section 5.1 followed by the experimental design for the task of predicting interests of users in Section 5.2.

5.1 Predicting Friendship Links

We have conducted a set of thirteen experiments designed to investigate the performance of several classifiers (listed above) at predicting friendship relationships when presented with different sets of features. When used alone, interest-based features may not be very effective at predicting friendships. However, one may expect the classifying algorithms to perform better when presented with graph features. Our results support this intuition (Chapter 6).

In each experiment, the training and test data sets are independent and each consists of 1000 user pairs. The training set contains approximately 50% friend pairs and 50% non-friend pairs, while the test set contains user pairs selected randomly from the original distribution (10%-15% friend pairs), under the assumption that two users are not friends if there is no direct link between them in the network graph i.e. the the network graph is "complete". In addition, any overlap in terms of users, between the training and test data sets, is removed to ensure the independence of the two sets.

We use both versions ("O1" and "O2") of the ontology (see Sections 3.3.1 and 3.3.2) in each experiment. Thus, we learn two classifiers for each experiment, one for each version of the ontology. Every experiment is repeated ten times by taking ten random training and test samples from the data. Results are reported as the average AUC and the associated standard deviation over ten repetitions. In addition, we compare the *contribution* of one version of the ontology vs. the other by performing paired t-tests for the two classifiers (to find out which one is better). We also perform the paired t-tests to compare the performance of the classifiers at the task of predicting friendships with ontology with the performance of the classifiers at that task without ontology.

A *paired t-test* can be used as a procedure for comparing the performances of two learning methods (or even two classifiers obtained with the same algorithm but with different data sets) using a fixed data set (training and test) [Mitchell, 1997, Chapter 5]. A classifier is trained according to each learning method on the training data and is tested on the test data set. Multiple repetitions (in general 10) of training and test are conducted using different

samples from the data to generate two sets of performance results, one for each classifier. These two sets of results are compared using the *paired t-test* method to investigate if one learning method is significantly better than the other.

As described in [Mitchell, 1997], within the context of *paired t-tests*, in order to compare the two learning methods the following two hypotheses are considered:

- Null hypothesis or H_0: On average, there is no difference between performances of the classifiers obtained with the two learning methods.
- Alternate hypothesis or H_a: On average, the performances of the classifiers obtained with the two learning methods are significantly different.

At a particular confidence level (95% in our case), a *paired t-test* value less than 0.05 indicates that the alternate hypothesis (H_a) holds true with a probability of $(1 - 0.05)$. Hence, we reject H_0 in favor of H_a and conclude that the performance of one classifier is statistically different than the performance of the other. However, if the outcome of the *paired t-test* is greater than 0.05, then H_0 cannot be rejected and based on the given sets of performance results, we cannot conclude if one learning approach is better than the other [Rice, 1998, Chapter 11].

In what follows, we describe the thirteen experiments performed to address the task of predicting friendship links using the interests' ontology:

1. Interest-based nominal features in absence of the ontology:

   In our first experiment, we consider the original interests of users and aim to investigate if they can be used to predict friendships. There are 22,000 interests, which give a set of about 44,000 possible interest-instances (as some interests have more than on descriptions). The input vector corresponding to each candidate pair comprises of 88,000 attributes (44,000 for each user in the pair) and the class "friendship". Given the large number of attributes, we do not expect the classifiers to perform well at predicting friendships.

2. Interest-based nominal features with the ontology:

   In our second experiment, the interests are refined according to the more abstract levels of the interests' hierarchy. As in the first experiment, a list of interests for each candidate user pair is presented to the prediction algorithms. Since the performance of classifiers is evaluated for each modified version of the data (each level of abstraction), this experiment also reveals the best level of abstraction in the interests' ontology. We perform this experiment with both versions of interests' hierarchy to investigate if features derived from one hierarchy are better predictors of friendships than features derived from the other. In the following Chapters, the experiment performed with ontology "O1" will be denoted as "Experiment 2(a)", and that performed with ontology "O2" will be denoted as "Experiment 2(b)".

3. Interest-based nominal features with sub-ontologies:

   The divisive step of the HAD algorithm (Section 3.2) provides four sub-ontologies namely *books' ontology*, *movies' ontology*, *words' ontology* and *phrases' ontology*, which combine together (as shown in Figure 3.6) to generate the ontology of interests. In the third experiment, we modify the original data by considering interests at different levels of abstraction in these sub-ontologies and compute interest-based nominal features from the modified data. In this experiment, we wish to find the "optimal" levels of abstraction in sub-ontologies for which classifying algorithms perform best when presented with interest-based nominal features. Specifically, when exploring a sub-ontology, interest-based nominal features are refined by varying the levels of abstraction in that sub-ontology while keeping levels of abstractions in other sub-ontologies fixed. The performances of classifiers at the task of prediction of friendship links, when presented with interest-based nominal features refined at different levels of abstraction, reveal the "best level" of abstraction in the sub-ontology considered. In a similar way, optimal levels of abstraction corresponding to each sub-ontology are obtained. Finally, interests are modified considering optimal levels in each sub-ontology,

and interest-based nominal features are obtained and are used to predict friendships between *Live Journal* users. This experiment is also performed for both versions of interests' hierarchy to investigate the effectiveness of one when compared with the other in predicting friendships among users. In the rest of the thesis, the experiment performed with sub-ontologies in the ontology "O1" will be denoted as "Experiment 3(a)", and that performed with sub-ontologies in the ontology "O2" will be denoted as "Experiment 3(b)".

4. Graph-based features:

   Our fourth experiment addresses the friend prediction problem by exploiting graph-based features. Hsu et al. [2006] have empirically shown that graph-based features are very effective in predicting friendship links in social networks. We, thus, expect the classifiers to perform well in this experiment.

5. Graph-based features and interest-based nominal features in absence of the ontology:

   The fifth experiment considers interest-based nominal features combined with graph-based features in absence of the ontology, and aims to explore possible improvements in the performance of classifications algorithms when presented with both types of features as compared to only one of the two types of features.

6. Graph-based features and interest-based nominal features with the ontology:

   The sixth experiment considers interest-based nominal features combined (as used in the second experiment) with graph-based features and aims to explore possible improvements in the performance of classifications algorithms when presented with both types of features as compared to only one of the two types of features in presence of the ontology. Like in "Experiment 2", results from this experiment also revel the "best" levels of abstraction in the interests' ontology, that should be used to refine interests when addressing the prediction problem using graph-based features and interest-based nominal features. The experiments performed using the ontology

"O1" and "O2" are denoted as "Experiment 6(a)" and Experiment "6(b)" respectively.

7. Graph-based features and interest-based nominal features with sub-ontologies:
   Our seventh experiment proceeds in a similar manner as experiment six. However, in this experiment we refine interests with respect to various levels of abstractions in sub-ontologies as done in "Experiment 3". We explore the sub-ontologies to find the "best" levels of abstraction in the same, at which interest-based nominal features should be refined to predict friendships in combination with graph-based features. In the rest of the thesis, the experiment performed with sub-ontologies in the ontology "O1" will be denoted as "Experiment 7(a)", and that performed with sub-ontologies in the ontology "O2" will be denoted as "Experiment 7(b)".

8. Interest-based numerical features in absence of the ontology:
   In our eight experiment, we use the interest-based numerical features proposed in [Aljandal et al., 2008]. These features are computed on the original data set *without* considering the concept hierarchies of interests.

9. Interest-based numerical features with the ontology:
   In the ninth experiment, we modify the original data by considering interests at different levels of abstraction in the interests' hierarchy and compute the interest-based numerical features from the modified data. The resulting features are then used to predict friendships among users. Intuitively, these classifiers are expected to show an improvement in performance, compared to the classifiers that use numerical features computed from the original data. Since the performance of classifiers is evaluated for each modified version of the data (each level of abstraction), this experiment also reveals the best level of abstraction in the interests' ontology. We denote this experiment as "Experiment 9(a)" and "Experiment 9(b)" when performed with ontology "O1" and "O2" respectively.

10. Interest-based numerical features with sub-ontologies:

Similar to "Experiment 9", this experiment addresses the task of predicting friendships using interest-based numerical features. However, in this experiment, we refine the interests by varying levels of abstraction in the sub-ontologies as done in "Experiment 3". The results observed reveal the best level of abstraction in sub-ontologies of the interests' hierarchy. This experiment is denoted as "Experiment 10(a)" and Experiment "10(b)" respectively when refining interest-based numerical features according to the ontology versions "O1" and "O2",

11. Graph-based features and interest-based numerical features in absence of the ontology:
    The eleventh experiment considers graph-based and interest-based numerical features derived without making use of the interests ontology. The classification algorithms are expected to show an improvement over the previous results for they are exposed to more information than the classifiers in those experiments.

12. Graph-based features and interest-based numerical features with the ontology:
    Like "Experiment 11", the twelfth experiment uses graph-based and interest-based numerical features to predict friendships. However,interest-based numerical features are derived using the ontology. And using this set of features, classification algorithms are expected to show improvements in performance when compared to results from previous experiment. The experiments performed using the ontology "O1" and "O2" are denoted as "Experiment 12(a)" and Experiment "12(b)" respectively.

13. Graph-based features and interest-based numerical features with sub-ontologies:
    This experiment is performed using the same set of features as "Experiment 12" except that interest-based numerical features are refined using sub-ontologies of the interests' hierarchy. The results from this experiment reveals the "best" levels of abstraction in sub-ontologies, according to which interests should be modified when addressing the prediction problem using a combination of interest-based numerical features and graph-based features. In the Chapters to follow, we denote the experiment conducted

using "O1" as "Experiment 13(a)" and that conducted using "O2" as "Experiment 13(b)".

5.2 Predicting Interests

We have conducted a set of three experiments designed to investigate the performance of several classifiers (listed in the beginning of this Chapter) at predicting interests of *Live Journal* users when presented with different sets of features. When used alone, interest-based nominal features may not be very effective at predicting interests. However, one may expect the classifying algorithms to perform better when presented with interest-based numerical features. Our results support this intuition (Chapter 6).

Given a set of k interests, we address the prediction problem at hand, by learning k classifiers, one for each interest. The data set consists of k interest-based nominal attributes (one for each interest) and/or 9 interest-based numerical attributes as described in Section 4.1. For each user, the task of each classifier is to predict whether or not the user has that interest for which the classifier is learned. For each of the k classifiers, we report the AUC averaged over 10 repetitions of the experiments. In the given data set, which consists of only 1000 users, we did not have enough positive examples (e.g. out of 1000 users, only 140 of them list "comedy books" as their interest) to experiment with separate training and test data sets. Hence, to perform 10 repetitions of the experiments we have used the "cross-validation" option in "Weka" data mining software.

The interests of each user can be represented as a vector of k boolean values $< a_1, a_2, a_3 ... a_k >$, where $a_i = 1$ if the user possess the i^{th} interest, or $a_i = 0$ otherwise. We call this vector as the "actual" interest vector. Similarly, user's predicted interests can also be represented as a vector of k boolean values $< p_1, p_2, p_3 ... p_k >$, where $p_i = 1$ if the i^{th} classifier predicts that the user possesses i^{th} interest, or $a_i = 0$ otherwise. We call this vector as the "predicted" interest vector. Hamming distance between the "actual" and the "predicted" interest vectors

is computed according to Equation 5.1.

$$hammingDistance(p, v) = \frac{1}{k} * \sum_{i=1}^{k} j \tag{5.1}$$

where$j = 1$ if $a_i = p_i$, and $j = 0$ otherwise.

By computing the hamming distance averaged over the number of users (here 1000), prediction performances of k classifiers are combined to report the overall performance of the classifiers at the task of interest prediction.

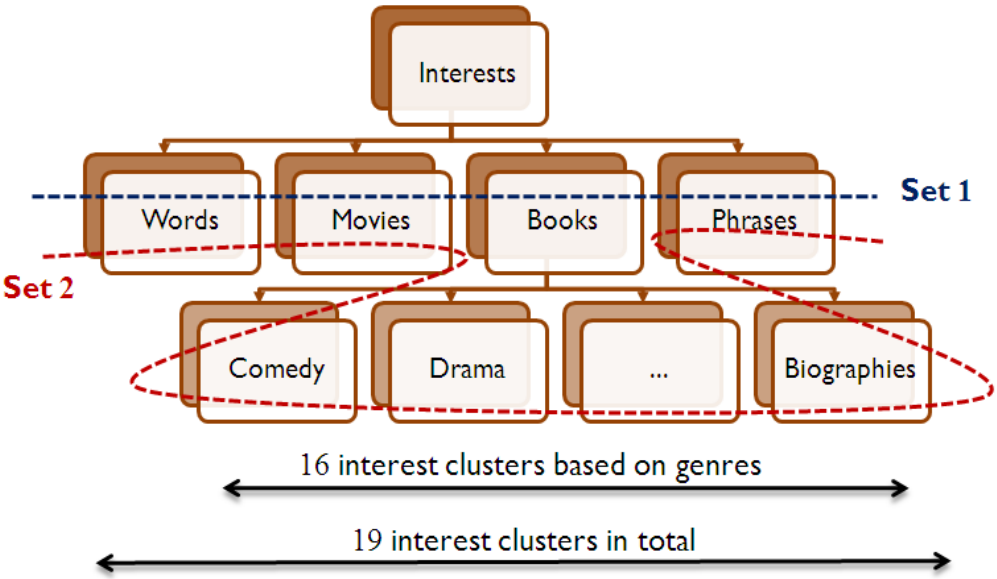

Figure 5.1: *Interests Considered for the Task of Prediction of Interests*

The number of examples in the data (1000 users) is too small to learn any classifier reasonably well, if original interests of the users ($k \approx 22,000$) are considered. In what follows, we address the problem of predicting interests by considering two sets of interests as shown in Figure 5.1. "Set 1" comprises of four interests namely *Books*, *Movies*, *Words* and *Phrases*, and "Set 2" comprises of nineteen interests namely *Movies*, *Words*, *Phrases* and sixteen genres into which *Books* are divided in the divisive clustering step of the HAD algorithm as described in Section 3.2. Future work, will address this problem with a larger data set (e.g. number of users $\geq 10,000$ and number of interests ≥ 19).

The set of three experiments performed to address the task of predicting interests of users are described as as follows:

1. Interest-based nominal features:

   In our first experiment, we present the classifiers with interest-based nominal features (as described in Section 4.1). To predict interests, we have 4 nominal features with respect to "Set 1" and 19 with respect to "Set 2". Interest-based nominal features are not as effective at capturing information from the data, as their numerical counterparts. With that said, we do not expect classifiers to perform well in this experiment.

2. Interest-based numerical features:

   In the second experiment, the input to the classifiers comprises of interest-based numerical features (as described in Section 4.1), and we expect to see an improvement in the performance of the classifiers when compared with their performances in the previous experiment.

3. Interest-based nominal and numerical features:

   In our third experiment, the input to the classifiers comprises of interest-based nominal features combined with interest-based numerical features. When presenting with both sets of features, we expect the classifiers to perform best when compared with their performances from the previous experiments.

We perform these experiments for both sets of interests, and report the AUC values and average hamming distances between "actual" and "predicted" interest vectors for each of them. A point worth noting is that interests from "Set 1" and "Set 2" exist at levels of abstraction 1 and 2 (see Figure 3.6). With respect to interests from the two sets, at levels 1 and 2, there is no distinction between two versions ("O1" and "O2") of the ontology. Hence, unlike experiments for predicting friendships, these experiments are not performed twice, one for each version of the ontology.

The results from both the prediction tasks are reported in Chapter 6 and conclusions derived from them are discussed in Chapter 8.

Chapter 6

Results

The main goal of the experiments described in Chapter 5 is to investigate the contribution of the interest ontology to the performance of algorithms for predicting friendships links and interests of *Live Journal* users. Indeed, the results indicate the usefulness of the ontology constructed. This Chapter is organized in two Sections; Section 6.1 presents the results corresponding the task of friendship prediction, and Section 6.2 shows the performances of classifiers at the task of predicting interests of users.

6.1 Predicting Friendship Links

In this section, we discuss the results of various classifiers at the task of predicting friendships between *Live Journal* users, with the aim to investigate the effectiveness of different types of features at that task. In addition, we explore the usefulness of considering various levels in individual sub-ontologies of the interests' hierarchy when computing interest-based features. We study the differences between the two versions versions of the interests' ontology (see Section 3.3) when using them to refine interest-based features and to predicting friendships.

We report the AUC values for classifiers at the task of predicting friendships between *Live Journal* users, averaged over ten repetitions of the experiments. As described in Section 5.1, we perform the paired t-tests to compare the performance of the classifiers at the task of predicting friendships with ontology with the performance of the classifiers at that task without ontology. We also perform paired t-tests to compare the contribution of one version

of the ontology vs. the other at that task. In addition, paired t-tests are also performed to compare the performance results of classifiers when presented with interest-based features refined using the "best" overall levels in the interests' ontology with their performance results when presented with interest-based features refined using the "best" levels in the sub-ontologies of the interests' hierarchy. We report t-values for all the paired t-tests conducted. A "t-value" less than 0.05 indicates the significance of the results at a 95% confidence level. **BOLD-FACED** AUC values in the Tables in this Chapter highlight the performance results of classifiers that are *statistically significantly* better than their respective baselines.

6.1.1 Interest-Based Nominal Features

This Section presents the performance results of classifiers at the task of predicting friendship links when presented with interest-based nominal features (Table 6.1). Nominal features aim at capturing the interest-based information in the data. However, the number of nominal features in the training and test data sets, sometimes, becomes too large for the classifiers to handle. E.g. for each user pair, $44,000$ user interests translate into $88,000$ interest-based nominal features. We have performed these experiments on machines with approximately 32 gigabytes of memory. When presented with nominal features in the absence of ontology, training and test sets become too large for Random Forest classifier to handle, and thus, experiments cannot be performed with the same (Experiment 1, Table 6.1). However, when refining nominal features in presence of the ontology, we are able to perform experiments and derive performance results (Experiments 2(a), 2(b) and 3(a), Table 6.1), except for Experiment 3(b) which is performed using interest-based nominal features with respect to best levels in sub-ontologies of the interests' hierarchy "O2". With respect to Experiment 3(b), dashes ("-") in Table 6.1 mean that the experiment did not finish within a time of 5 days on a machine with 32 gigabytes of memory.

We consider the AUC values from Experiment 1 (predicting friendships using interest-based nominal features in the absence of interests' hierarchy), as the *baseline* for comparing

Table 6.1: *AUC for different classifiers presented with interest-based nominal features*

Exp#	Ontology	SVM	Logistic	J48	RandomForest	OneR
1		0.65± 0.11	0.64±0.10	0.60±0.09	-	0.56±0.08
2(a)	O1	*0.74±0.13*	*0.68±0.10*	**0.70±0.12**	0.77±0.12	0.53±0.06
2(b)	O2	0.64±0.078	0.63±0.09	*0.62±0.09*	0.69±0.05	0.55± 0.06
3(a)	Sub-O1	**0.74±0.09**	0.62 ±0.07	**0.62±0.07**	0.65±0.03	0.56±0.05
3(b)	Sub-O2	-	-	-	-	-

Table 6.2: *Paired t-tests*

Exp#	SVM	Logistic	J48	RandomForest	OneR
1 vs. 2(a)	0.61	0.218	**0.02**	-	0.204
1 vs. 2(b)	0.379	0.467	0.271	-	0.443
2(a) vs. 2(b)	**0.021**	0.153	0.051	0.06	0.211
1 vs. 3(a)	**0.028**	0.224	**0.02**	-	0.457
1 vs. 3(b)	-	-	-	-	-
3(a) vs. 3(b)	-	-	-	-	-
2(a) vs. 3(a)	0.44	0.06	**0.029**	**0.006**	0.12
2(b) vs. 3(b)	-	-	-	-	-

the performances of classifiers when presented with nominal features that are refined using the interests' hierarchy. **BOLD-FACED** AUC values highlight the performances of the classifiers that are *statistically significantly* better than the baseline.

Table 6.1 shows that, when friendships are predicted using the interest-based nominal features refined using best overall levels in ontology "O1", three out of five classifiers (SVM, Logistic Regression and J48) show an improvement in the performance when compared with the baseline (Experiments 1 and 2(a)). However, paired t-test values suggest that performance improvement of only J48 classifier is statistically significant at 95% level of confidence. Furthermore, when using ontology "O2", only J48 shows an improvement in the performance at that task compared with the baseline, but, this improvement too is not statistically significant.

Table 6.1 also presents the AUC values for classifiers at the task of predicting friendships using interest-based nominal features refined according to "best" levels in the sub-ontologies of the interests' hierarchy "O1"(Experiments 3(a)). We observe that, in this case only two

out of five classifiers (SVM and OneR) show significant improvement, compared to the baseline.

Moreover, comparing AUC values for Experiments 2(a) with 3(a) from Table 6.1, it becomes clear that three out of five classifiers (Logistic Regression, J48 and Random Forest) perform better when presented with interest-based nominal features refined using "best" overall levels in the interests' ontology "O1", than when presented with interest-based nominal features refined using "best" levels in the sub-ontologies of the same. However, according to paired t-test values, the significance of these observations is only demonstrated for J48 and Random Forest classifiers.

Paired t-test values for the pair of Experiments 2(a) vs. 2(b) are also presented in Table 6.2 to investigate if one version of the ontology performs better at predicting friendships (given interest-based nominal features only) than the other version. As observed from the Table 6.2, t-test values for all classifiers except SVM are greater than 0.05 suggesting only SVM performs better at the task of predicting friendships when presented with nominal features refined in presence of the ontology "O1", compared with its performance at that task when presented with the same set of features refined in presence of the ontology "O2". For all other cases, we cannot conclude if nominal features refined in the presence of one ontology are more effective at predicting friendships in *Live Journal* than the other.

Table 6.3: *Best levels of abstraction in ontologies for predicting friendships using interest-based nominal features*

Ontology	# of Levels	Best Level
Ontology O1	18	6
Ontology O2	44	7
Books Ontology O1	16	12
Books Ontology O2	30	-
Movies Ontology O1	15	8
Movies Ontology O2	43	-
Words Ontology O1	17	9
Words Ontology O2	31	-
Phrases Ontology O1	14	7
Phrases Ontology O2	20	-

Table 6.3 shows the best levels of abstraction (other than the root representing "everything" and the leaves representing the original interests) in the overall interests' ontology and in the sub-ontologies, when addressing the prediction of friendships using interest-based nominal features only. Results related to these experiments are shown in Chapter 10 (Appendix A). Please note that in the experimental results, when the ontology is used to construct features and compare the performance of different classifiers, the ontology is considered from the respective best levels of abstraction.

6.1.2 Graph-Based Features and Interest-Based Nominal Features

This Section presents the performance results of classifiers at the task of predicting friendship links when presented with graph-based features and interest-based nominal features (Table 6.4). As mentioned in Section 6.1.2, graph-based features alone are very effective at predicting friends in the given social network (Experiment 4, Table 6.4). When presented with both graph-based and interest-based nominal features, the classifiers are expected to show improvements in their prediction performances. However, results from the Table 6.4 show otherwise. Only one out of five classifiers (OneR) shows the expected improvement in the performance at that task when presented with a combination of graph-based features and interest-based nominal features in the absence of ontology (Experiment 4 vs. 5, Table 6.4).

Furthermore, as mentioned in Section 6.1.1, the number of nominal features in the training and test data sets, sometimes, becomes too large for the classifiers to handle. In this case too, we have performed these experiments on machines with approximately 32 gigabytes of memory. When presented with graph-based features and nominal features in the absence of ontology, the number of features become too overwhelming for Random Forest classifier to handle, and thus, experiments cannot be performed with the same (Experiment 5, Table 6.4). However, when refining nominal features in presence of the ontology, we are able to perform experiments and derive performance results (Experiments 6(a), 6(b) and 7(a),

Table 6.4), except for Experiment 7(b) which is performed using graph-based features and interest-based nominal features with respect to best levels in sub-ontologies of the interests' hierarchy "O2". With respect to Experiment 7(b), dashes ("-") in Table 6.4 mean that the experiment did not finish within a time of 5 days on a machine with 32 gigabytes of memory.

Table 6.4: *AUC for different classifiers presented with graph-based features and interest-based nominal features*

Exp#	Ontology	SVM	Logistic	J48	RandomForest	OneR
4		0.92±0.03	0.91±0.04	0.94±0.03	0.97±0.03	0.86±0.09
5		0.89±0.06	0.84±0.06	0.89±0.05	-	0.88±0.03
6(a)	O1	*0.91±0.05*	0.76±0.07	*0.91±0.05*	0.82±0.08	0.88±0.04
6(b)	O2	*0.92± 0.03*	0.75±0.05	*0.91±0.05*	0.86±0.07	0.88±0.07
7(a)	Sub-O1	0.88±0.061	0.85±0.07	0.92±0.04	0.88±0.04	**0.92±0.05**
7(b)	Sub-O2	-	-	-	-	-

Table 6.5: *Paired t-tests*

Exp#	SVM	Logistic	J48	RandomForest	OneR
5 vs. 6(a)	0.299	**0.021**	0.368	-	0.367
5 vs. 6(b)	0.08	**0.001**	0.44	-	0.362
6(a) vs. 6(b)	0.144	0.15	0.454	0.094	0.454
5 vs. 7(a)	0.316	0.451	0.368	-	**0.027**
5 vs. 7(b)	-	-	-	-	-
7(a) vs. 7(b)	-	-	-	-	-
6(a) vs. 7(a)	0.137	**0.023**	0.366	**0.026**	0.07
6(b) vs. 7(b)	-	-	-	-	-

We consider the AUC values from Experiment 5, i.e. predicting friendships using graph-based features and interest-based nominal features in the absence of interests' hierarchy, as the *baseline* for comparing the performances of classifiers when presented with the same set of features refined using the interests' hierarchy. **BOLD-FACED** AUC values highlight the performances of the classifiers that are *statistically significantly* better than the baseline.

Table 6.4 shows that, when friendships are predicted using graph-based and interest-based nominal features refined using best overall levels in ontology "O1" and "O2", two out of five classifiers (SVM and J48) show an improvement in the performance when compared

with the baseline (Experiments 5, 6(a) and 6(b)). However, paired t-test values suggest that the performance improvements are not statistically significant at 95% level of confidence.

Table 6.4 also presents the AUC values for classifiers at the task of predicting friendships using graph-based and interest-based nominal features refined according to "best" levels in the sub-ontologies of the interests' hierarchy "O1"(Experiments 7(a)). We observe that, in this case only OneR classifier shows statistically significant improvement, compared to the baseline.

Moreover, comparing AUC values for Experiments 6(a) with 7(a) from Table 6.4, it becomes clear that four out of five classifiers (Logistic Regression, J48, Random Forest and OneR) perform better when presented with graph-based and interest-based nominal features refined using "best" levels in sub-ontologies of the interests' ontology "O1", than when presented with the same set of features refined using "best" overall levels in the same ontology. However, according to paired t-test values, the significance of these observations is only demonstrated for Logistic Regression and Random Forest classifiers.

Paired t-test values for the pair of Experiments 6(a) vs. 6(b) are also presented in Table 6.5 to investigate if one version of the ontology performs better at predicting friendships (given graph-based and interest-based nominal features) than the other version. As observed from the Table 6.5, t-test values for all classifiers are greater than 0.05 suggesting that we cannot conclude if one ontology performs better than the other (Experiments 6(a) and 6(b)).

Table 6.6 shows the best levels of abstraction (other than the root representing "everything" and the leaves representing the original interests) in the overall interests' ontology and in the sub-ontologies, when addressing the prediction of friendships using graph-based features and interest-based nominal features. Results related to these experiments are shown in Chapter 11 (Appendix B). Please note that in the experimental results, when the ontology is used to construct features and compare the performance of different classifiers, the ontology is considered from the respective best levels of abstraction.

Table 6.6: *Best levels of abstraction in ontologies for predicting friendships using graph-based features and interest-based nominal features*

Ontology	# of Levels	Best Level
Ontology O1	18	7
Ontology O2	44	6
Books Ontology O1	16	13
Books Ontology O2	30	-
Movies Ontology O1	15	3
Movies Ontology O2	43	-
Words Ontology O1	17	3
Words Ontology O2	31	-
Phrases Ontology O1	14	5
Phrases Ontology O2	20	-

6.1.3 Interest-Based Numerical Features

In this Section, we present the performance results of classifiers at the task of predicting friendship links when presented with interest-based numerical features (Table 6.7). Numerical features aim at capturing the interest-based information in the data, and at the same time ensure that the number of features in the training and test data sets is not too large for the classifiers to handle. The results in Table 6.7 show an improvement in the performance of classifiers, when interest-based numerical features are used as opposed to the interests themselves (Experiments 1 vs. Experiment 8), thus suggesting that interest-based numerical features are better at summarizing interest-based information than their nominal counterparts.

Table 6.7: *AUC for different classifiers presented with interest-based numerical features*

Exp#	Ontology	SVM	Logistic	J48	RandomForest	OneR
1						
8		0.66±0.07	0.64±0.08	0.59±0.09	0.61±0.09	0.58±0.09
9(a)	O1	**0.74±0.08**	**0.73±0.10**	**0.68±0.08**	*0.66±0.09*	**0.67±0.07**
9(b)	O2	**0.76±0.11**	**0.73±0.12**	**0.69±0.09**	**0.73±0.11**	**0.64±0.09**
10(a)	Sub-O1	0.61±0.21	0.62±0.19	*0.60±0.17*	0.60±0.14	*0.59±0.09*
10(b)	Sub-O2	0.62±0.12	0.62±0.16	*0.60±0.09*	0.58±0.13	0.55±0.08

We consider the AUC values from Experiment 8 (predicting friendships using interest-

Table 6.8: *Paired t-tests*

Exp#	SVM	Logistic	J48	RandomForest	OneR
8 vs. 9(a)	**0.017**	**0.029**	**0.016**	0.121	**0.021**
8 vs. 9(b)	**0.024**	**0.04**	**0.015**	**0.008**	**0.008**
9(a) vs. 9(b)	0.382	0.46	0.44	0.068	0.265
8 vs. 10(a)	0.257	0.406	0.451	0.426	0.466
8 vs. 10(b)	0.173	0.372	0.378	0.329	0.216
10(a) vs. 10(b)	0.447	0.483	0.465	0.423	0.193
9(a) vs. 10(a)	**0.049**	0.066	0.09	0.142	**0.025**
9(b) vs. 10(b)	**0.009**	**0.05**	**0.03**	**0.008**	**0.017**

based numerical features in the absence of interests' hierarchy), as the *baseline* for comparing the performances of classifiers when presented with numerical features that are refined using the interests' hierarchy. **BOLD-FACED** AUC values highlight the performances of the classifiers that are *statistically significantly* better than the baseline.

Table 6.7 shows that, when friendships are predicted using the interest-based numerical features refined using "best" overall levels in either versions of the ontology (O1 or O2), all classifiers show an improvement in the performance when compared with the baseline (Experiments 8, 9(a) and 9(b)). When using ontology "O1", the performance improvements of four out of five classifiers are significant at 95% level of confidence, and when using ontology "O2", the performance improvements of all the classifiers are significant at that level.

Table 6.7 also presents the AUC values for classifiers at the task of predicting friendships using interest-based numerical features refined according to "best" levels in the sub-ontologies of the interests' hierarchy (Experiments 10(a) and 10(b)). The values *italicized* highlight the performances of the classifiers that show an improvement in comparison with the baseline. However, these improvements are not significant statistically, at the confidence level of 95%.

Moreover, comparing AUC values for Experiments 9(a) with 10(a) and that of 9(b) with 10(b) from Table 6.7, it becomes clear that all the classifiers perform better when presented with interest-based numerical features refined using "best" overall levels in the interests'

ontology, than when presented with interest-based numerical features refined using "best" levels in the sub-ontologies of the interests' hierarchy. Paired t-test values suggest that these observations for interests' ontology "O2" are statistically significant for all classifiers. However, for interests' ontology "O1", the significance of these observations is only demonstrated for SVM and OneR classifiers.

Paired t-test values for pairs of Experiments 9(a) vs. 9(b), and 10(a) vs. 10(b) are also presented in Table 6.8 to investigate if one version of the ontology performs better at predicting friendships (given interest-based numerical features only) than the other version. As observed from the Table 6.8, t-test values for all classifiers are greater than 0.05 suggesting that we cannot conclude if one ontology performs better than the other, when interest-based numerical features are modified with respect to "best" overall levels in the ontology (Experiments 9(a) and 9(b)) or "best" levels in the sub-ontologies (Experiments 10(a) and 10(b)).

Table 6.9: *Best levels of abstraction in ontologies for predicting friendships using interest-based numerical features*

Ontology	# of Levels	Best Level
Ontology O1	18	14
Ontology O2	44	28
Books Ontology O1	16	8
Books Ontology O2	30	23
Movies Ontology O1	15	8
Movies Ontology O2	43	31
Words Ontology O1	17	14
Words Ontology O2	31	13
Phrases Ontology O1	14	7
Phrases Ontology O2	20	3

Table 6.9 shows the best levels of abstraction (other than the root representing "everything" and the leaves representing the original interests) in the overall interests' ontology and in the sub-ontologies, when addressing the prediction of friendships using interest-based numerical features only. Results related to these experiments are shown in Chapter 12 (Appendix C). Please note that in the experimental results, when the ontology is used

to construct features and compare the performance of different classifiers, the ontology is considered from the respective best levels of abstraction.

6.1.4 Graph-Based Features and Interest-Based Numerical Features

This section focuses on the performance results of classifiers at the task of predicting friendship links when presented with graph-based features and interest-based numerical features. The AUC values of the classifiers are shown in Table 6.10. As mentioned in Section 6.1.2, graph-based features alone are very effective at predicting friends in the given social network. When presented with both graph-based and interest-based numerical features, the classifiers are expected to show improvements in their prediction performances. Table 6.10 indicates that three out of five classifiers (SVM, Logistic Regression and Random Forest classifiers) show the expected improvement (Experiment 4 vs. Experiment 11). Thus, for some classifiers, graph-based features combined with interest-based numerical features result in better performance compared to graph-based features alone.

Table 6.10: *AUC for different classifiers presented with graph-based features and interest-based numerical features*

Exp	Ontology	SVM	Logistic	J48	RandomForest	OneR
4		0.92±0.03	0.91±0.04	0.94±0.03	0.97±0.03	0.86±0.09
11		0.92±0.03	0.91±0.04	0.94±0.02	0.98±0.01	0.86±0.09
12(a)	O1	*0.94±0.04*	*0.94±0.02*	0.93±0.05	0.97±0.02	*0.88±0.04*
12(b)	O2	**0.95±0.03**	*0.94±0.03*	0.94±0.03	0.98±0.01	**0.91±0.04**
13(a)	Sub-O1	0.90±0.05	0.91±0.035	0.94±0.03	0.97±0.03	0.86±0.06
13(b)	Sub-O2	*0.93±0.04*	*0.92±0.04*	0.93±0.05	0.98±0.01	*0.91±0.08*

We consider the AUC values from Experiment 11 i.e predicting friendships using graph-based and interest-based numerical features in the absence of interests' hierarchy, as the *baseline* for comparing the performances of classifiers when presented with the same set of features in the presence of interests' hierarchy. **BOLD-FACED** AUC values highlight the performances of the classifiers that are *statistically significantly* better than the baseline.

Table 6.11: *Paired t-tests*

Experiments	SVM	Logistic	J48	RandomForest	OneR
11 vs. 12(a)	0.209	0.067	0.268	0.311	0.297
11 vs. 12(b)	**0.035**	0.082	0.466	0.401	**0.039**
12(a) vs. 12(b)	0.199	0.46	0.314	0.249	0.025
11 vs. 13(a)	0.148	0.325	0.359	0.346	0.444
11 vs. 13(b)	0.393	0.33	0.341	0.305	0.162
13(a) vs. 13(b)	0.128	0.169	0.278	0.237	0.099
12(a) vs. 13(a)	**0.05**	**0.015**	0.219	0.493	0.187
12(b) vs. 13(b)	0.105	0.153	0.33	0.402	0.288

Table 6.10 shows that, when friendships are predicted using the graph-based and interest-based numerical features refined using "best" overall levels in either versions of the ontology (O1 or O2), SVM, Logistic and OneR classifiers show an improvement in the performance when compared with the baseline (Experiments 11, 12(a) and 12(b)). When using ontology "O2", SVM and OneR classifiers show significant improvements compared with the baseline. However, the paired t-test values suggest that performance improvements in the classifiers when using "O1" are not significant at 95% confidence level. Thus, we cannot conclude if the performance of the classifiers when using "O1", is better than the baseline.

Table 6.10 also presents the AUC values for classifiers at the task of predicting friendships using graph-based and interest-based numerical features refined according to "best" levels in the sub-ontologies of the interests' hierarchy (Experiments 13(a) and 13(b)). The *italicized* values highlight the performances of the classifiers that show an improvement in comparison with the baseline. However, these improvements are not statistically significant, at the confidence level of 95%.

Moreover, comparing AUC values for Experiments 12(a) with 13(a) and that of 12(b) with 13(b) from Table 6.10, it becomes clear that all classifiers, except Random Forest, perform better when presented with graph-based and interest-based numerical features refined using "best" overall levels in the interests' ontology, than when presented with graph-based features and interest-based numerical features refined using "best" levels in the sub-ontologies of the interests' hierarchy. Random Forest classifier performs equally well in both

cases. Table 6.11 shows that, for second version (O2) of the interests' ontology, significance of these observations cannot be demonstrated (($t > 0.05$, Experiment 12(b) vs. 13(b)). However, for the first version (O1), $t < 0.05$ for "SVM" and Logistic Regression classifiers indicating that, the observations made above are statistically significant for these two classifiers at a 95% confidence level (Experiment 12(a) vs. 13(a)).

Paired t-test values for pairs of Experiments 12(a) vs. 12(b), and 13(a) vs. 13(b) are also presented in Table 6.11 to investigate if one version of the ontology performs better at predicting friendships (given graph-based features and interest-based numerical features) than the other version. As observed from the Table 6.11, t-test values for all classifiers are greater than 0.05 suggesting that we cannot conclude if one ontology performs better than the other when interest-based numerical features are modified with respect to "best" overall levels (Experiments 12(a) and 12(b)) or "best" levels in the sub-ontologies (Experiments 13(a) and 13(b)) of the interests' hierarchy.

Table 6.12: *Best Levels of abstraction in ontologies for predicting friendships using graph-based features and interest-based numerical features*

Ontology	# of Levels	Best Level
Ontology O1	18	7
Ontology O2	44	37
Books Ontology O1	16	13
Books Ontology O2	30	8
Movies Ontology O1	15	13
Movies Ontology O2	43	39
Words Ontology O1	17	5
Words Ontology O2	31	12
Phrases Ontology O1	14	11
Phrases Ontology O2	20	18

Table 6.12 shows best levels of abstraction (other than the root representing "everything" and the leaves representing the original interests) in the overall interests' ontology and in the sub-ontologies, when addressing the prediction of friendships using graph-based features and interest-based numerical features. Results related to these experiments are shown in Chapter 13 (Appendix D). Please note that in the experimental results, when the ontology is

used to construct features and compare the performance of different classifiers, the ontology is considered from the respective best levels of abstraction.

6.2 Predicting Interests

As mentioned in Section 5.2, the task of predicting interests of users of the *Live Journal* social network is addressed using two sets of interests. "Set 1" comprises of four interests while "Set 2" constitutes of nineteen interests (see Figure 5.1). This section presents the performance results of the classifying algorithms at predicting interests from these two sets. For each experiment, we report the hamming distances (see Section 5.2) averaged over the number of users (1000) to present accuracy of the classifiers at the task of predicting interests. In addition, the AUC values for the classifiers for predicting each individual interest are reported in Chapter 14 (Appendix E).

Table 6.13: *Average hamming distance for different classifiers when predicting interests from "Set 1"*

Exp#	Nominal	Numerical	SVM	Logistic	J48	RandomForest	OneR
1	*		0.792	0.793	0.793	1	0.789
2		*	0.991	0.989	0.99	0.989	0.988
3	*	*	0.991	0.989	0.99	0.989	0.988

Table 6.13 shows average hamming distances for classifiers at the task of predicting interests from "Set 1". As we can observe, for all of the classifiers except Random Forest, interest-based nominal features (Experiment 1) are not effective at predicting interests (hamming distance ≈ 0.79). However, Random Forest classifier performs perfectly at that task. Random Forest classifier consists of many decision trees. For each instance, it predicts the class that is the mode of classes predicted by individual trees given that instance [Mitchell, 1997]. In Experiment 1, our data set consists of four interest-based nominal attributes (one for each interest). Since our classifiers are subjected to the task of predicting four interests, our conjecture is that some of the individual trees constituting the Random Forest classifier may have observed "overfitting" to generate perfect results .

Hamming distances from "Experiment 2" (Table 6.13) show that all classifiers perform well (hamming distance $\approx$ 0.99) when presented with interest-based numerical features. Furthermore, the performance of these classifiers does not improve when presented with a combination of interest-based nominal and numerical features. From results shown in Table 6.13, we can conclude that interest-based numerical features alone are are effective at summarizing interest information to predict interests from "Set 1". However, given that every user has some interest from "Set 1", our training data is very unbalanced towards the positive examples, and thus, "almost" perfect results in "Experiment 2" and "Experiment 3" are not surprising.

Table 6.14: *Average hamming distance for different classifiers when predicting interests from "Set 2"*

Exp#	Nominal	Numerical	SVM	Logistic	J48	RandomForest	OneR
1	*		0.69	0.69	0.688	0.68	0.68
2		*	0.867	0.871	0.858	0.842	0.851
3	*	*	0.869	0.87	0.854	0.853	0.851

Table 6.14 shows average hamming distances for classifiers at the task of predicting interests from "Set 2". Results from "Experiment 1" show that interest-based nominal features are not effective at predicting interests (hamming distance for each classifier $\approx$ 0.69). When presented with interest-based numerical features, all classifiers show about 25% improvement in their prediction performance. Furthermore, the performance of two out of five classifiers (SVM and Random Forest) improves when presented with a combination of interest-based nominal and numerical features (Experiment 3, Table 6.14). However, when presented with the same combination, J48 classifier shows a slight decline in its performance compared to its performance when presented with numerical features alone.

Table 6.15 shows the best classifier (**BOLD-FACED** values) for predicting interests from "Set 2", given a particular set of features. As we can observe, Logistic Regression classifier performs the best in all cases. Future work may consider using the same when predicting interests for larger data sets.

Table 6.15: *Best classifier for predicting interests from "Set 2"*

Exp#	Nominal	Numerical	SVM	Logistic	J48	RandomForest	OneR
1	*		**0.69**	**0.69**	0.688	0.68	0.68
2		*	0.867	**0.871**	0.858	0.842	0.851
3	*	*	0.869	**0.87**	0.854	0.853	0.851

Table 6.16: *Best set of features for predicting interests from "Set 2"*

Exp#	Nominal	Numerical	SVM	Logistic	J48	RandomForest	OneR
1	*		0.69	0.69	0.688	0.68	0.68
2		*	0.867	**0.871**	**0.858**	0.842	**0.851**
3	*	*	**0.869**	0.87	0.854	**0.853**	**0.851**

Table 6.16 shows the set of features (**BOLD-FACED** values) for predicting interests from "Set 2", given a particular classifier. Two out of five classifiers (SVM and Random Forest) perform best when presented with the combination of interest-based numerical and nominal features. OneR classifier performs equally well when presented with numerical features alone and when presented with both numerical and nominal features. While the other two classifiers (Logistic and OneR) perform best when presented with interest-based numerical features alone. Results from Table 6.16 suggest that, both interest-based numerical and a combination of interest-based numerical and nominal features are effective at predicting interests from "Set 2". Depending on the choice of classifying algorithm used, one may prefer the former set of features over the latter or vice versa.

Chapter 7

Related Work and Discussion

This Chapter provides a short review of the areas of research related to the work presented in the paper. Section 7.1 presents the progress in the field of semantic information extraction in social networks using simple clustering-based approaches. Section 7.2 describes recent advancements in the field of social network analysis.

7.1 Information Extraction in Social Networks

"The success and popularity of social network systems have generated many interesting and challenging problems to the research community" [Li et al., 2008]. Li et al. [Li et al., 2008] have addressed the problem of discovering social interests shared by a group of users of the social network system "del.icio.us". Their approach is based on the key observation that "human users tend to use descriptive tags to bookmark or annotate the web-based content they are interested in". A system called "Internet Social Interest Discovery" (ISID) that clusters user-generated tags, bookmarked URLs and users who annotated the URLs with these tags is described in [Li et al., 2008] and used to predict user interests. The results show that ISID successfully discovers interests of nearly 90% users. Similar to the ISID system presented by Li et al., our approach uses clustering techniques to group user interests together. In *Live Journal*, interests are analogous to bookmarked URLs in "del.icio.us" and are explicitly specified by the users. Furthermore, "user-tags" in "del.icio.us" are analogous to descriptions of interests extracted by HAD from various sources, in our approach. However,

HAD does not cluster users and their interests together, and uses the semantic information captured in the ontology to address the problem of predicting friend relationships.

In other related work, Mori et al. [Mori et al., 2006] have investigated the use of semantic information extracted from web documents to discover relationships in social networks. Specifically, given entity pairs in a political social network (e.g., George W. Bush - United States, Junichiro Koizumi - Japan), the goal is to extract labels for describing the relations of the respective entity pairs (i.e., to discover relevant terms that relate a politician to a location in this example). Their approach first builds a context model for each entity pair, based on the web based contents that describe the participating entities. The algorithm, then, clusters entity pairs according to the similarities among the corresponding context models and finally completes by selecting representative labels to describe relations from each cluster. With reference to our approach, entity-pairs are analogous to interests whose descriptions (context models) can be extracted from the information available on the web. Moreover, interests are clustered based on their descriptions to capture the underlying semantics. Unlike Mori et al., however, we use the resulting clusters (ontology) to predict only one kind of relationship in Live Journal, i.e. "Friendships".

7.2 Social Network Analysis

"Social network discovery" refers to the general task that comprises of specific subtasks of analysis (prediction) and visualization (exploration) of social networks. This section provides information on the current research and accomplished work related to analyzing and visualizing social networks.

Broadly speaking, analyzing a social network is a challenging problem in itself. In addition, it largely depends on the quality of data given to address the challenge. A common approach to improving data quality is "Entity Resolution" [Bilgic et al., 2006]. Bilgic et al. [Bilgic et al., 2006] have developed an interactive tool "D-Dupe", which provides a stable visual layout for optimized entity resolution and allows users to combine entity resolution

algorithms including data mining algorithms for entity resolution and task-specific network visualization. As the paper describes, the entity resolution process can be seen as an iterative process: as pairs of nodes are resolved, additional duplicates may be revealed; therefore, resolution decisions are often chained together. D-Dupe users resolve ambiguities either by merging nodes or by marking them distinct. In addition they can apply sequences of actions to produce a high quality entity resolution result [Bilgic et al., 2006].

In terms of social network analysis, two commonly addressed, but independently studied problems are: object classification (labeling the nodes of a graph) and link prediction (predicting the links in a graph). Object classification is performed assuming a complete set of known links. Coffman and Marcus [Coffman and Marcus, 2003] have addressed the task of object classification by characterizing actors in a simulated dataset as terrorists or non-terrorists by applying statistical classifiers to their social network analysis (SNA) metric values. As the case study describes, the simulated datasets modeled the social interactions that occur within Leninist cell organizations as well as in more typical social structures. Furthermore, the authors achieve an accuracy of 86% in a three-class classification problem (cell leader, cell member, or non-terrorist) and an accuracy of 96% in a two-class classification problem (terrorist or non-terrorist). On the other hand, problems such as link-based classification, identifying the link type, predicting the link strength and cardinality, that come under the umbrella of link prediction are addressed assuming a fully observed set of node attributes [Getoor, 2003]. Hsu et al. [Hsu et al., 2007] have addressed the problems of predicting, classifying, and annotating friendship relations in a social network, based on the network structure and user profile data. In their approach, all the node attributes are available from the blog service *Live Journal*. They address a set of link prediction tasks such as predicting existing links and estimating inter-pair distance and achieve an accuracy of about 98%.

In most real world domains, however, attributes and links are often missing or incorrect. Object classification is not provided with all the links relevant to correct classification and

link prediction is not provided with all the labels needed for accurate link prediction [Bilgic et al., 2007]. Bilgic [Bilgic et al., 2007] have developed an approach that addresses these two problems by interleaving object classification and link prediction in a "simple yet general" collective classification algorithm. In their approach they provide the object prediction task with information about the available node and its links. Once a prediction about a node or its particular attribute is made, it is used to address the prediction problem for related links and in general for other links in the network as well. The results show that the algorithm performs well (nearly 90% accuracy) when compared to "flat" prediction approach (addressing each problem independently) [Bilgic et al., 2007].

The second subtask of social network discovery i.e. visual mining of the underlying network has also received much attention in the recent research on social networks. Singh et al. [Singh et al., 2007] have developed a tool "Invenio" which makes use of a wide range of interactive visualization features included in "Prefuse" [pre, 2007] the open source tool kit for graph visualization, graph mining algorithm support from JUNG and construction of views from both database and graph-based operations. With "Invenio" they aim to explore the multi-modal multi relational social networks. As they describe, "modal" refers to number of node-sets in N and "relational" refers to the number of relationship types in N, where N is an extended multi-modal multi-relational network. Besides the features provided by "prefuse" [pre, 2007], "Invenio" provides newly developed features such as attribute guided subgraph generation, graph mining and visual application analysis.

Chapter 8

Conclusion

Live Journal is an online journal service with an emphasis on user interaction [Fitzpatrick, 1999]. Among other things in the *Live Journal* social network, users can list their interests and tag other users as their friends. Given the emphasis on user interaction, *Live Journal* network can be represented by a graph structure, wherein nodes of the graph represent the users of the journal service and edges represent friendship links. The tasks of predicting friendships between users and interests of *Live Journal* users come under the umbrella of Social Network Analysis.

In social networks, these prediction problems can be addressed by using a variety of features, e.g. interest-based features and graph-based features. Hsu et al. [2007] have addressed the task of predicting friendship links using these features. Their results suggest that features constructed only from the common interests of two users are ineffective at predicting friend relationships. While network graph features prove to give good results, several questions can be still raised: Can we improve the performance of the algorithms that use graph features by combining them with interest-based features? If we are interested in other prediction problems, such as the prediction of the users' interests themselves, how can we handle the overwhelming number of interests? In this thesis, we have addressed the problem of building an interests' ontology and predicting potential friendship relations between users in the social network Live Journal, using features constructed based on the interests' ontology. Our goal has been to organize users' interests in an ontology (specifically, a concept

hierarchy) and to use the semantics captured by this ontology to improve the performance of learning algorithms at predicting if two users can be friends. In addition, we have also addressed the problem of predicting interests of users using the features constructed from the interests' ontology.

The HAD algorithm, proposed in this thesis, constructs a concept hierarchy of interests of users, making it possible for classification algorithms to use semantic information in the form of an ontology. The algorithm is implemented by combining hierarchical agglomerative and divisive clustering approaches to overcome the shortcomings of other related approaches. Two flavors of the agglomerative phase resulting in two versions of the interests' ontology are explored. Furthermore, the algorithm implementation is general and can be used to construct similar ontologies in other domains.

We have performed experiments to explore the effectiveness of interest-based numerical and nominal features at predicting friendships when used in presence of the ontology. Our results show that interest-based numerical features perform better at capturing interest-information (and hence predicting friendships) than their nominal counterparts. Our investigation also shows that both versions of the ontology are very useful in predicting friendships using interest-based numerical features refined with respect to "best" overall levels in the ontologies. However, experiments which investigate the effectiveness of interest-based numerical features, refined with respect to "best" levels in sub-ontologies of the interests' hierarchy, do not show consistent improvements when predicting friendships.

Graph-based features are very effective at predicting friendships in the *Live Journal* social network [Hsu et al., 2007]. Results from our experiments show that performance of classifiers does not improve when using the combination of graph-based features and interest-based nominal features in the presence of ontology, compared to their performance at that task when presented with graph-based features alone. Furthermore, our investigations show that the combination of graph-based and interest-based numerical features derived in the presence of the ontology are effective at this problem in some cases. Specifically,

two out of five classifiers (SVM and OneR) show statistically significant improvements in the performance of predicting friendships when presented with graph-based features and interest-based numerical features refined with respect to the "best" overall level in second version (O2) of the interests' ontology. In addition, classifiers using graph-based features and interest-based numerical features refined with respect to "best" levels in the sub-ontologies do not show an improvement in their prediction performance compared to the classifiers using graph-based features combined with interest-based numerical features refined with respect to "best" overall levels in the interests' ontology. In addition, for all sets of features, results from our experiments (specially paired t-tests) suggest that between the ontologies "O1" and "O2", neither one is better at predicting friendships than the other.

For the task of predicting interests of users, the number of examples in the available data (1000 users) is too small to learn any classifier reasonably well, if original interests of the users ($\approx 22,000$) are considered. In this thesis, we have addressed the problem by considering two sets of interests. "Set 1" comprises of four interests namely *Books*, *Movies*, *Words* and *Phrases*, and "Set 2" comprises of nineteen interests namely *Movies*, *Words*, *Phrases* and sixteen genres into which *Books* are divided in the divisive clustering step of the HAD algorithm.

Results from our experiments show that while interest-based nominal features are not as effective at predicting interests, interest-based numerical features and a combination of the two are very effective at that task. All classifiers perform "almost" perfect at predicting interests from "Set 1" when presented the two latter sets of features. However, given that every user has some interest from "Set 1", our training data is very unbalanced towards the positive examples, and thus, "almost" perfect results are not surprising. With that said, it is the performance results of classifiers at the task of predicting interests from "Set 2", that show the effectiveness of numerical features and the combination of numerical and nominal features at that task. Our investigations reveal that all classifiers show an improvement of at least 25% when predicting interests using numerical features compared to predicting

interests using nominal features alone. Two out of five classifiers (SVM and Random Forest) further show slight improvements at that task when presented with a combination of the two features. Thus, depending on the choice of the classification algorithm used, one may prefer the numerical features over combination of both numerical and nominal features for predicting interests, or vice versa.

This work has contributed to significant advances to the state of the art in Social Network Mining by organizing (without any human interaction) the interests of users into an interests' ontology, so that semantic knowledge can be incorporated into interest-based features, thus improving the performance of classifiers trained to predict friendships and interests of users in social networks.

Chapter 9

Future Work

This chapter showcases several related problems that we would like to address in future work. They are briefly described in what follows:

- Predicting friendship links in incomplete social networks:

  As we mentioned in Chapter 5, our experiments are based on the assumption that *Live Journal* social network is "complete", i.e. if there is no link between nodes "A" and "B" in the social network graph then, *Live Journal* users "A" and "B" are not friends in real life as well. However, a social network may be incomplete, meaning that if two users "A" and "B" are friends in real life, but they have not tagged each other as friends in *Live Journal*, then links between nodes "A" and "B" will be missing from the graph. And thus, the graph features so derived will be incomplete. An extension to our work may address the problem of predicting friendship links in such incomplete social networks possibly via considering approaches such as time-series learning. In time-series learning, data from *Live Journal* is collected over a series time-points and graph-based features are refined to capture the length of time for which a friendship exists. In addition, interest-based features can also be refined to incorporate the length of time for which a user is interested in an interest etc. And thus, by using the features so derived, friendships in incomplete social network can be estimated as a function of time.

- Including user-defined ontologies in Ontology Engineering:

  Similar to the *Live Journal* social network, other social networks such as "Flickr","Facebook" or "del.icio.us" can be considered. Specifically, these social networks allow users to tag their interests, resulting in small "user defined" ontologies [Syed et al., 2008]. One idea for future work is to build the ontology of interests on top of these user-defined ontologies. Moreover, like the ontology of interests, ontologies of schools and communities can be derived from *Live Journal* and included in the ontology of interests.

- Incorporating information from Wikipedia and Google:

  In future, our work can be extended to use Wikipedia's topic ontology to engineer concept hierarchy of interests [Syed et al., 2008]. Wikipedia is a freely available online encyclopedia developed by a community of users. The documents or articles in Wikipedia represent a consensus view of the users on the topics the articles describe, and are linked to other articles which describe related concepts [Syed et al., 2008]. Wikipedia's articles and the underlying article link graph form a *topic ontology* that is developed and maintained by the Wikipedia community [Syed et al., 2008]. Topics in the topic ontology can be seen as interests and thus, articles describing these topics as interest descriptions. In addition, interests can be queried to search engines, e.g. *Google* and *MSN Live Search*, and the results so retrieved can also be considered as articles describing the interests [Gupta and Oates, 2007]. Given a set of articles, interests can be compared with one another using techniques such as *Latent Semantic Analysis* [Deerwester et al., 1990] to guide the grouping of related interests into clusters (representing concepts), and hence into a concept hierarchy. Furthermore, our interest ontology only represents "is-a" relationships between concepts in a hierarchical manner. With that said, articles describing more than one interest and Wikipedia's article link graph may also be used to derive other relationships such as "a-kind-of","has-a","hyponym-of", "subsumes", "subtype-of" etc, to enrich our ontology to represent more than one relationship.

- Using other similarity measures:

  In Section 3.3, we mention that similarity between two interests (and hence, two interest-clusters) is computed as the dot product between vectors representing these interests according to a boolean representation. Future work will look at developing a controlled term vocabulary based on all interest definitions, and hence representing interests as weighted vectors of terms in the vocabulary [Manning et al., 2008, Chapters 2,6]. Similarities between two interests (and hence, two interest-clusters) will then be computed as the dot product or cosine between two vectors representing the interests, as described in the implementation of *Vector Space Model* for information retrieval [Manning et al., 2008, Chapter 6]. Furthermore, semantics in the controlled vocabulary of terms may also be elevated, using *Latent Semantic Indexing*, to combine terms and give improved measures of similarity between two interests[Deerwester et al., 1990; Manning et al., 2008, Chapter 18].

- Refining interest-based measures using a variable-depth cut through the ontologies:

  Chapter 5 delineates the use of interest ontologies in refining numerical interest-based measures. Specifically, interests are viewed at various levels of abstraction in the ontologies and numerical interest-based measures are computed as described in Section 4.2. Instead of refining interest-based measures using fixed levels of abstraction in the ontologies, the notion of refining the same using a variable depth cut through the ontologies can be considered in future. Following the idea presented by Zhang et al. [2005, Section 3.1], the ontologies can be greedily explored by beginning the refinement of interest-based measures using the most abstract cut through the concept hierarchy. At each cluster node in the hierarchy, a selection can be made to further refine interest-based measures by a) considering interests constituting this node or b) considering its child clusters, with the aim to optimize the performance of classifiers in predicting friendship links in *Live Journal* social network. Following this approach, one could refine interest-based measures using a *global cut* [Zhang et al., 2005] through

the ontologies such that the performance of classifiers in predicting friendships is maximized.

- Combining the two versions of ontology of interests:

  Section 3.3 presents two versions of hierarchical agglomerative step of the HAD algorithm (Chapter 3), which generate two versions of the ontology of interests. Future work extending this thesis, can also address the problem of combining these two versions and investigating the effect of the same in performance of classifying algorithms at predicting friendship links and/or interests of users of *Live Journal*.

Bibliography

Prefuse: interactive information visualization toolkit, 2007. Released on October 21st 2007.

W. Aljandal, W. H. Hsu, V. Bahirwani, D. Caragea, and T. Weninger. Validation-based normalization and selection of interestingness measures for association rules. In *Proceedings of Artificial Neural Networks in Engineering, (ANNIE)*, 2008.

V. Bahirwani, D. Caragea, W. Aljandal, and W. H. Hsu. "ontology engineering and feature construction for predicting friendship links in the live journal social network. In *Proceedings of the 2nd ACM Workshop on Social Network Mining and Analysis, (SNA-KDD)*, 2008.

P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002. URL http://citeseer.nj.nec.com/berkhin02survey.html.

M. Bilgic, L. Licamele, L. Getoor, and B. Shneiderman. D-dupe: an interactive tool for entity resolution in social networks. In *Visual Analytics Science and Technology, (VAST)*, Baltimore, MD, USA, October 2006.

M. Bilgic, G. M. Namata, and L. Getoor. Combining collective classification and link prediction. In *Workshop on Mining Graphs and Complex Structures at the IEEE International Conference on Data Mining, (ICDM)*, 2007.

P. Buitelaar, P. Cimiano, and B. Magnini. Ontology learning from text: an overview. IOS Press, 2003.

D. Calvanese, G. D. Giacomo, M. Lenzerini, and M. Y. Vardi. View-based query processing: on the relationship between rewriting, answering and losslessness. In *Proceedings of the*

10th International Conference on Database Theory (ICDT), volume 3363 of *LNCS*, pages 321–336. Springer, 2005.

D. Caragea, A. Silvescu, J. Pathak, J. Bao, C. Andorf, D. Dobbs, and V. Honavar. Information integration and knowledge acquisition from semantically heterogeneous biological data sources. In *Proceedings of the Second International Workshop on Data Integration in Life Sciences, (DILS)*, San Diego, CA, 2005. Berlin: Springer-Verlag. Lecture Notes in Computer Science.

T. R. Coffman and S. E. Marcus. Pattern classification in social network analysis: a case study. In *Aerospace Conference*, volume 5, pages 3162–3175, 2003.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. "indexing by latent semantic analysis. In *Journal of the American Society for Information Science*, volume 41, 1990.

A. Doan and A. Halevy. Semantic integration research in the database community: a brief survey. *AI Magazine, Special Issue on Semantic Integration*, 26(1):83–94, 2005. URL http://anhai.cs.uiuc.edu/home/papers/si-survey-db-community.pdf.

B. Eckman. A practitioner's guide to data management and data integration in bioinformatics. *Bioinformatics*, pages 3–74, 2003.

T. Fawcett. An introduction to roc analysis. In *Pattern Recognition Letters*, 2005.

B. Fitzpatrick. Live journal: online journal service, 1999. Created in 1999.

L. Geng and H. J. Hamilton. Interestingness measures for data mining : a survey. *ACM Computer Survey*, 2006.

L. Getoor. Link mining: a new data mining challenge. *SIGKDD Explorations*, 5(1):85–89, 2003.

M. Godoy and A. Amandi. Modeling user interests by conceptual clustering. http://www.sciencedirect.com, 2005.

T. R. Gruber. A translation approach to portable ontology specifications. Technical Report 5(2):199-220, Knowledge Systems AI Laboratory, Stanford University, April 1993.

T. R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In *International Journal of Human and Computer Studies*, volume 43, pages 907–928, 1994.

A. Gupta and T. Oates. Using ontologies and the web to learn lexical semantics. In *Proceedings of the International Joint Conference on Artificial Intelligence, (IJCAI)*, 2007.

H. Happel and S. Seedorf. Application of ontologies in software engineering. In *Semantic Web Enabled Software Engineering*, 2006.

A. Hotho, S. Steffen, and G. Stumme. Ontologies improve text document clustering. In *Proceedings of The Third IEEE International Conference on Data Mining*, 2003.

W. H. Hsu, A. L. King, M. S. R. Paradesi, T. Pydimarri, and T. Weninger. Collaborative and structural recommendation of friends using weblog-based social network analysis. In *Proceedings of Computational Approaches to Analyzing Weblogs, (AAAI)*, 2006.

W. H. Hsu, J. Lancaster, M. S. R. Paradesi, and T. Weninger. Structural link analysis from user profiles and friends networks: a feature construction approach. In *Proceedings of the International Conference on Weblogs and Social Media, (ICWSM)*, Boulder, CO, USA, March 2007.

W. H. Hsu, T. Weninger, and M. S. R. Paradesi. Predicting links and link change in friends networks: supervised time series learning with imbalanced data. In *Proceedings of Artificial Neural Networks in Engineering, (ANNIE)*, 2008.

A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999. URL citeseer.ist.psu.edu/jain99data.html.

D. K. Kang, A. Silvescu, J. Zhang, and V. Honavar. Generation of attribute value taxonomies from data for data-driven construction of accurate and compact classifiers. In *Proceedings of the 4th IEEE International Conference on Data Mining, (ICDM)*, 2004.

Hyoung R. Kim and Philip K. Chan. Learning implicit user interest hierarchy for context in personalization. In *Proceedings of the 8th International Conference on Intelligent User Interfaces, (IUI)*, pages 101–108, New York, NY, USA, 2003. ACM. ISBN 1-58113-586-6. doi: http://doi.acm.org/10.1145/604045.604064.

A. Levy. Logic-based techniques in data integration. In *Logic-based artificial intelligence*, pages 575–595. Kluwer Academic Publishers, 2000.

X. Li, L. Guo, and Y. Zhao. Tag-based social interest discovery. In *Proceedings of the 17th International World Wide Web Conference, (WWW)*, Beijing, China, 2008.

C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to information retrieval*. Cambridge University Press, first edition, 2008.

D. L. McGuinness. Ontologies come of age. In Dieter Fensel, Jim Hendler, Henry Lieberman, and Wolfgang Wahlster, editors, *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2003.

T. M. Mitchell. *Machine learning*. McGraw-Hill Companies Inc., international edition, 1997.

J. Mori, T. Tsujishita, Y. Matsuo, and M. Ishizuka. Extracting relations in social networks from the web using similarity between collective contexts. In *Proceedings of the 5th International Semantic Web Conference, (ISWC)*, Athens, Georgia, 2006.

N. Noy and H. Stuckenschmidt. Ontology alignment: an annotated bibliography. In Y. Kalfoglou, M. Schorlemmer, A. Sheth, S. Staab, and M. Uschold, editors, *Semantic Interoperability and Integration*, number 04391 in Dagstuhl Seminar Proceedings, 2005.

K. Punera, S. Rajan, and J. Ghosh. Automatic construction of n-ary tree based taxonomies. In *Proceedings of Ontology Mining and Knowledge Discovery from Semistructured Documents Workshop, (MSD) at ICDM*, 2006.

J. A. Rice. *Mathematical statistics and data analysis*. Wadsworth and Brooks/Cole Advanced Books and Software, tenth edition, 1998.

P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. In *Genome Research*, pages 2498–2504, 2003.

B. Shridharan, A. Tretiakov, and Kinshuk. Application of ontology to knowledge management in web based learning. In *IEEE International Conference on Advanced Learning Technologies*, 2004.

L. Singh, M. Beard, L. Getoor, and M. B. Blake. Visual mining of multi-modal social networks at different abstraction levelsx. In *L. Singh, M. Beard, L. Getoor, M. Blake. Visual mining of multi-modal social networks at different abstraction levels. IEEE Conference on Information Visualization - Symposium of Visual Data Mining (IV-VDM)*, 2007.

N. Slonim and N. Tishby. Agglomerative information bottleneck. In *Proceedings of the 13th Neural Information Processing Systems, (NIPS)*, 1999.

Z. S. Syed, T. Finin, and A. Joshi. Wikipedia as an ontology for describing documents. In *Proceedings of the 2nd International Conference on Weblogs and Social Media, (ICWSM)*, 2008.

P. N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proceeding of the ACM SIGKDD International Conference on Knowledge Discovery in Databases, (KDD)*, pages 32–41. ACM SIGKDD, 2002.

I. H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, and S. J. Cunningham. Weka: practical machine learning tools and techniques with java implementations. In Nikola Kasabov and Kitty Ko, editors, *Proceedings of the ICONIP/ANZIIS/ANNES'99 Workshop on Emerging Knowledge Engineering and Connectionist-Based Information Systems*, pages 192–196, 1999. Dunedin, New Zealand.

J. Zhang, D. Caragea, and V. Honavar. Learning ontology-aware classifiers. In *Proceedings of the Eight International Conference on Discovery Science, (DS)*, volume 3735, pages 308–321. Berlin: Springer-Verlag, 2005.

Chapter 10

Appendix A

In this Chapter, we report the AUC values for "Support Vector Machine" classifier (SVM) at the task of predicting friendships among *Live Journal* users, when presented with interest-based nominal features in presence of the interests' ontology. The AUC values are averaged over ten repetitions of each experiment. We also report the standard deviation associated with the average AUC values in each case. Interest-based features are refined by varying the levels of abstraction in the ontology and the sub-ontologies, and for each experiment, the "best" levels of abstraction (i.e. the level of abstraction for which SVM performs the best) are highlighted using **BOLD-FACED** font. The choice of the classifier (SVM) comes from the fact that, preliminary experiments of our work [Bahirwani et al., 2008] suggest that SVM is the best classifier for predicting friendships in Live Journal using interest-based features alone, or a combination of graph features and interest-based features in the presence of interests' ontology.

The Tables 10.1 and 10.2 show the performance of SVM classifiers at predicting friendships when presented with interest-based nominal features refined using "best" overall levels in the interests' ontologies "O1" and "O2" respectively. In addition, Tables 10.3 to 10.6 show the performance of SVM at that task when presented with same set of features refined using "best" levels in sub-ontologies of the interests' hierarchy "O1".

Table 10.1: *AUC for SVM presented with interest-based nominal features refined using ontology "O1"*

Level#	AUC	Level#	AUC
1	0.70±0.16	9	0.70±0.10
2	0.63±0.10	10	0.64±0.12
3	0.70±0.09	11	0.62±0.09
4	0.70±0.09	12	0.64±0.09
5	0.67±0.11	13	0.63±0.16
6	**0.75±0.13**	14	0.68±0.12
7	0.69±0.17	15	0.69±0.08
8	0.67±0.15	16	0.65±0.11

Table 10.2: *AUC for SVM presented with interest-based nominal features refined using ontology "O2"*

Level#	AUC	Level#	AUC
1	0.66±0.13	22	0.69±0.11
2	0.67±0.13	23	0.64±0.09
3	0.68±0.10	24	0.67±0.11
4	0.64±0.09	25	0.71±0.13
5	0.71±0.12	26	0.59±0.08
6	0.70±0.12	27	0.62±0.10
7	**0.74±0.11**	28	0.65±0.13
8	0.70±0.16	29	0.60±0.13
9	0.72±0.08	30	0.61±0.14
10	0.66±0.15	31	0.64±0.12
11	0.68±0.13	32	0.61±0.17
12	0.64±0.08	33	0.69±0.06
13	0.62±0.08	34	0.73±0.09
14	0.72±0.12	35	0.58±0.11
15	0.63±0.10	36	0.63±0.08
16	0.64±0.06	37	0.67±0.15
17	0.64±0.10	38	0.63±0.09
18	0.66±0.13	39	0.60±0.10
19	0.64±0.10	40	0.68±0.13
20	0.70±0.09	41	0.68±0.11
21	0.68±0.10	42	0.64±0.09

Table 10.3: *AUC for SVM presented with interest-based nominal features refined using books ontology "O1"*

Level#	AUC	Level#	AUC
1	0.68±0.10	8	0.76±0.08
2	0.64±0.11	9	0.73±0.10
3	0.69±0.10	10	0.63±0.10
4	0.69±0.08	11	0.67±0.09
5	0.67±0.12	**12**	**0.76±0.08**
6	0.63±0.09	13	0.70±0.14
7	0.65±0.10	14	0.66±0.16

Table 10.4: *AUC for SVM presented with interest-based nominal features refined using movies ontology "O1"*

Level#	AUC	Level#	AUC
1	0.65±0.17	**8**	**0.76±0.08**
2	0.67±0.12	9	0.62±0.09
3	0.70±0.10	10	0.68±0.08
4	0.63±0.11	11	0.65±0.10
5	0.69±0.07	12	0.67±0.10
6	0.63±0.12	13	0.65±0.08
7	0.62±0.14		

Table 10.5: *AUC for SVM presented with interest-based nominal features refined using words ontology "O1"*

Level#	AUC	Level#	AUC
1	0.66±0.13	**9**	**0.76±0.08**
2	0.60±0.11	10	0.63±0.09
3	0.65±0.14	11	0.66±0.11
4	0.62±0.12	12	0.64±0.09
5	0.71±0.09	13	0.60±0.19
6	0.60±0.12	14	0.62±0.13
7	0.69± 0.11	15	0.65±0.07
8	0.67±0.06		

Table 10.6: *AUC for SVM presented with interest-based nominal features refined using phrases ontology "O1"*

Level#	AUC	Level#	AUC
1	0.69±0.10	**7**	**0.76±0.08**
2	0.68±0.11	8	0.74±0.08
3	0.73±0.09	9	0.64±0.06
4	0.70±0.12	10	0.64±0.10
5	0.74±0.07	11	0.65±0.15
6	0.71±0.09	12	0.68±0.10

Chapter 11

Appendix B

In this Chapter, we report the AUC values for "Support Vector Machine" classifier (SVM) at the task of predicting friendships among *Live Journal* users, when presented with a combination of graph-based features and interest-based nominal features in presence of the interests' ontology. The AUC values are averaged over ten repetitions of each experiment. We also report the standard deviation associated with the average AUC values in each case. Interest-based features are refined by varying the levels of abstraction in the ontology and the sub-ontologies, and for each experiment, the "best" levels of abstraction (i.e. the level of abstraction for which SVM performs the best) are highlighted using **BOLD-FACED** font. The choice of the classifier (SVM) comes from the fact that, preliminary experiments of our work [Bahirwani et al., 2008] suggest that SVM is the best classifier for predicting friendships in Live Journal using interest-based features alone, or a combination of graph features and interest-based features in the presence of interests' ontology.

The Tables 11.1 and 11.2 show the performance of SVM classifiers at predicting friendships when presented with graph-based features and interest-based nominal features refined using "best" overall levels in the interests' ontologies "O1" and "O2" respectively. In addition, Tables 11.3 to 11.6 show the performance of SVM at that task when presented with same set of features refined using "best" levels in sub-ontologies of the interests' hierarchy "O1".

Table 11.1: *AUC for SVM presented with graph-based features and interest-based nominal features refined using ontology "O1"*

Level#	AUC	Level#	AUC
1	0.92±0.05	9	0.90±0.06
2	0.89±0.05	10	0.89±0.05
3	0.90±0.05	11	0.88±0.04
4	0.89±0.07	12	0.90±0.05
5	0.89±0.07	13	0.86±0.11
6	0.90±0.07	14	0.89±0.07
7	**0.91±0.05**	15	0.87±0.08
8	0.85±0.15	16	0.89±0.06

Table 11.2: *AUC for SVM presented with graph-based features and interest-based nominal features refined using ontology "O2"*

Level#	AUC	Level#	AUC
1	0.86±0.10	22	0.88±0.07
2	0.85±0.08	23	0.82±0.04
3	0.87±0.06	24	0.86±0.06
4	0.89±0.06	25	0.87±0.08
5	0.92±0.05	26	0.82±0.09
6	**0.92±0.03**	27	0.84±0.06
7	0.91±0.07	28	0.81±0.08
8	0.92±0.05	29	0.81±0.12
9	0.92±0.05	30	0.88±0.06
10	0.87±0.08	31	0.86±0.07
11	0.90±0.07	32	0.87±0.10
12	0.87±0.07	33	0.88±0.05
13	0.89±0.05	34	0.89±0.05
14	0.91±0.05	35	0.87±0.08
15	0.87±0.08	36	0.82±0.07
16	0.86±0.04	37	0.89±0.10
17	0.88±0.06	38	0.85±0.09
18	0.83±0.11	39	0.80±0.08
19	0.83±0.09	40	0.87±0.04
20	0.89±0.06	41	0.89±0.07
21	0.84±0.07	42	0.84±0.05

Table 11.3: *AUC for SVM presented with graph-based features and interest-based nominal features refined using books ontology "O1"*

Level#	AUC	Level#	AUC
1	0.91±0.04	8	0.90±0.05
2	0.90±0.04	9	0.88±0.07
3	0.89±0.05	10	0.83±0.12
4	0.89±0.09	11	0.90±0.06
5	0.87±0.05	12	0.89±0.05
6	0.89±0.04	**13**	**0.91±0.05**
7	0.86±0.06	14	0.90±0.06

Table 11.4: *AUC for SVM presented with graph-based features and interest-based nominal features refined using movies ontology "O1"*

Level#	AUC	Level#	AUC
1	0.88±0.07	8	0.90±0.05
2	0.85±0.07	9	0.90±0.03
3	**0.91±0.03**	10	0.89±0.04
4	0.84±0.06	11	0.86±0.08
5	0.88±0.04	12	0.88±0.09
6	0.87±0.05	13	0.89±0.05
7	0.88±0.06		

Table 11.5: *AUC for SVM presented with graph-based features and interest-based nominal features refined using words ontology "O1"*

Level#	AUC	Level#	AUC
1	0.89±0.04	9	0.90±0.05
2	0.83±0.04	10	0.86±0.06
3	**0.90±0.07**	11	0.87±0.07
4	0.86±0.09	12	0.85±0.09
5	0.90±0.05	13	0.77±0.13
6	0.87±0.05	14	0.86±0.10
7	0.87±0.05	15	0.87±0.05
8	0.89±0.03		

Table 11.6: *AUC for SVM presented with graph-based features and interest-based nominal features refined using phrases ontology "O1"*

Level#	AUC	Level#	AUC
1	0.89±0.05	7	0.90±0.05
2	0.88±0.06	8	0.88±0.05
3	0.88±0.05	9	0.86±0.06
4	0.88±0.06	10	0.87±0.06
5	**0.91±0.04**	11	0.87±0.05
6	0.88±0.06	12	0.86±0.05

Chapter 12

Appendix C

In this Chapter, we report the AUC values for "Support Vector Machine" classifier (SVM) at the task of predicting friendships among *Live Journal* users, when presented with interest-based numerical features in presence of the interests' ontology. The AUC values are averaged over ten repetitions of each experiment. We also report the standard deviation associated with the average AUC values in each case. Interest-based features are refined by varying the levels of abstraction in the ontology and the sub-ontologies, and for each experiment, the "best" levels of abstraction (i.e. the level of abstraction for which SVM performs the best) are highlighted using **BOLD-FACED** font. The choice of the classifier (SVM) comes from the fact that, preliminary experiments of our work [Bahirwani et al., 2008] suggest that SVM is the best classifier for predicting friendships in Live Journal using interest-based features alone, or a combination of graph features and interest-based features in the presence of interests' ontology.

The Tables 12.1 and 12.2 show the performance of SVM classifiers at predicting friendships when presented with interest-based numerical features refined using "best" overall levels in the interests' ontologies "O1" and "O2" respectively. In addition, Tables 12.3 to 12.10 show the performance of SVM at that task when presented with same set of features refined using "best" levels in sub-ontologies of the interests' hierarchies "O1" and "O2".

Table 12.1: *AUC for SVM presented with interest-based numerical features refined using ontology "O1"*

Level#	AUC	Level#	AUC
1	0.62±0.15	9	0.65±0.09
2	0.61±0.1	10	0.62±0.11
3	0.58±0.1	11	0.65±0.12
4	0.61±0.12	12	0.67±0.1
5	0.6±0.13	13	0.65±0.12
6	0.62±0.13	**14**	**0.74±0.08**
7	0.64±0.12	15	0.66±0.07
8	0.65±0.13	16	0.67±0.13

Table 12.2: *AUC for SVM presented with interest-based numerical features refined using ontology "O2"*

Level#	AUC	Level#	AUC
1	0.59±0.13	22	0.70±0.14
2	0.57±0.11	23	0.63±0.14
3	0.62±0.10	24	0.62±0.13
4	0.56±0.12	25	0.70±0.13
5	0.63±0.10	26	0.67±0.11
6	0.67±0.14	27	0.74±0.10
7	0.62±0.10	**28**	**0.76±0.11**
8	0.68±0.15	29	0.73±0.11
9	0.61±0.10	30	0.64±0.15
10	0.64±0.10	31	0.68±0.08
11	0.60±0.20	32	0.63±0.14
12	0.64±0.07	33	0.69±0.12
13	0.64±0.10	34	0.68±0.08
14	0.62±0.07	35	0.69±0.13
15	0.58±0.14	36	0.72±0.12
16	0.69±0.11	37	0.68±0.10
17	0.74±0.10	38	0.69±0.09
18	0.64±0.14	39	0.68±0.09
19	0.71±0.11	40	0.73±0.13
20	0.71±0.12	41	0.68±0.13
21	0.61±0.11	42	0.67±0.10

Table 12.3: *AUC for SVM presented with interest-based numerical features refined using books ontology "O1"*

Level#	AUC	Level#	AUC
1	0.65±0.11	**8**	**0.68±0.09**
2	0.51±0.12	9	0.60±0.12
3	0.60±0.10	10	0.55±0.09
4	0.66±0.12	11	0.65±0.12
5	0.65±0.11	12	0.67±0.15
6	0.60±0.14	13	0.66±0.09
7	0.63±0.08	14	0.58±0.09

Table 12.4: *AUC for SVM presented with interest-based numerical features refined using books ontology "O2"*

Level#	AUC	Level#	AUC
1	0.64±0.18	15	0.70±0.10
2	0.64±0.10	16	0.72±0.06
3	0.59±0.14	17	0.66±0.09
4	0.57±0.09	18	0.70±0.11
5	0.65±0.12	19	0.61±0.08
6	0.63±0.17	20	0.66±0.08
7	0.63±0.11	21	0.62±0.12
8	0.69±0.13	22	0.68±0.09
9	0.58±0.11	**23**	**0.78±0.07**
10	0.59±0.11	24	0.64±0.12
11	0.64±0.16	25	0.68±0.11
12	0.66±0.14	26	0.66±0.12
13	0.58±0.10	27	0.66±0.12
14	0.64±0.10	28	0.66±0.13

Table 12.5: *AUC for SVM presented with interest-based numerical features refined using movies ontology "O1"*

Level#	AUC	Level#	AUC
1	0.65±0.14	**8**	**0.68±0.09**
2	0.62±0.12	9	0.62±0.11
3	0.64±0.12	10	0.65±0.10
4	0.59±0.10	11	0.62±0.16
5	0.60±0.10	12	0.61±0.06
6	0.65±0.08	13	0.62±0.08
7	0.64±0.06		

Table 12.6: *AUC for SVM presented with interest-based numerical features refined using movies ontology "O2"*

Level#	AUC	Level#	AUC
1	0.61±0.16	22	0.70±0.10
2	0.52±0.10	23	0.67±0.08
3	0.56±0.13	24	0.66±0.12
4	0.60±0.17	25	0.69±0.10
5	0.66±0.11	26	0.70±0.10
6	0.63±0.13	27	0.69±0.15
7	0.66±0.13	28	0.64±0.09
8	0.63±0.11	29	0.62±0.15
9	0.67±0.05	30	0.66±0.07
10	0.60±0.12	31	0.76±0.10
11	0.61±0.11	32	0.61±0.12
12	0.66±0.11	33	0.61±0.12
13	0.66±0.11	34	0.62±0.17
14	0.64±0.11	35	0.68±0.15
15	0.74±0.09	36	0.69±0.10
16	0.66±0.13	37	0.69±0.14
17	0.61±0.16	38	0.62±0.14
18	0.70±0.10	39	0.71±0.09
19	0.65±0.08	40	0.64±0.11
20	0.69±0.13	41	0.75±0.11
21	0.67±0.09		

Table 12.7: *AUC for SVM presented with interest-based numerical features refined using words ontology "O1"*

Level#	AUC	Level#	AUC
1	0.59±0.19	9	0.68±0.09
2	0.60±0.07	10	0.58±0.10
3	0.61±0.11	11	0.58±0.17
4	0.63±0.16	12	0.58±0.11
5	0.65±0.12	13	0.63±0.12
6	0.63±0.11	**14**	**0.69±0.09**
7	0.65±0.08	15	0.64±0.10
8	0.66±0.08		

Table 12.8: *AUC for SVM presented with interest-based numerical features refined using words ontology "O2"*

Level#	AUC	Level#	AUC
1	0.60±0.14	16	0.70±0.10
2	0.57±0.15	17	0.73±0.13
3	0.56±0.21	18	0.66±0.13
4	0.53±0.10	19	0.69±0.14
5	0.61±0.09	20	0.64±0.09
6	0.56±0.13	21	0.72±0.10
7	0.62±0.10	22	0.69±0.12
8	0.64±0.09	23	0.73±0.12
9	0.62±0.10	24	0.70±0.09
10	0.66±0.12	25	0.71±0.08
11	0.64±0.12	26	0.70±0.14
12	0.55±0.21	27	0.68±0.12
13	**0.75±0.09**	28	0.70±0.08
14	0.62±0.13	29	0.70±0.07
15	0.65±0.10		

Table 12.9: *AUC for SVM presented with interest-based numerical features refined using phrases ontology "O1"*

Level#	AUC	Level#	AUC
1	0.61±0.11	**7**	**0.68±0.09**
2	0.67±0.10	8	0.64±0.08
3	0.63±0.11	9	0.65±0.13
4	0.62±0.13	10	0.63±0.09
5	0.55±0.17	11	0.61±0.06
6	0.62±0.13	12	0.65±0.10

Table 12.10: *AUC for SVM presented with interest-based numerical features refined using phrases ontology "O2"*

Level#	AUC	Level#	AUC
1	0.62±0.10	10	0.70±0.10
2	0.57±0.19	11	0.62±0.10
3	**0.74**±0.09	12	0.66±0.11
4	0.58±0.09	13	0.68±0.10
5	0.66±0.09	14	0.72±0.14
6	0.67±0.12	15	0.68±0.13
7	0.63±0.07	16	0.65±0.10
8	0.68±0.09	17	0.69±0.09
9	0.61±0.11	18	0.68±0.23

Chapter 13

Appendix D

In this Chapter, we report the AUC values for "Support Vector Machine" classifier (SVM) at the task of predicting friendships among *Live Journal* users, when presented with a combination of graph-based features and interest-based numerical features in presence of the interests' ontology. The AUC values are averaged over ten repetitions of each experiment. We also report the standard deviation associated with the average AUC values in each case. Interest-based features are refined by varying the levels of abstraction in the ontology and the sub-ontologies, and for each experiment, the "best" levels of abstraction (i.e. the level of abstraction for which SVM performs the best) are highlighted using **BOLD-FACED** font. The choice of the classifier (SVM) comes from the fact that, preliminary experiments of our work [Bahirwani et al., 2008] suggest that SVM is the best classifier for predicting friendships in Live Journal using interest-based features alone, or a combination of graph features and interest-based features in the presence of interests' ontology.

The Tables 13.1 and 13.2 show the performance of SVM classifiers at predicting friendships when presented with graph-based features and interest-based numerical features refined using "best" overall levels in the interests' ontologies "O1" and "O2" respectively. In addition, Tables 13.3 to 13.10 show the performance of SVM at that task when presented with same set of features refined using "best" levels in sub-ontologies of the interests' hierarchies "O1" and "O2".

Table 13.1: *AUC for SVM presented with graph-based features and interest-based numerical features refined using ontology "O1"*

Level#	AUC	Level#	AUC
1	0.92±0.04	9	0.91±0.06
2	0.88±0.09	10	0.90±0.05
3	0.89±0.04	11	0.92±0.02
4	0.89±0.06	12	0.93±0.03
5	0.91±0.04	13	0.92±0.04
6	0.93±0.03	14	0.93±0.04
7	**0.94±0.04**	15	0.91±0.05
8	0.89±0.11	16	0.92±0.05

Table 13.2: *AUC for SVM presented with graph-based features and interest-based numerical features refined using ontology "O2"*

Level#	AUC	Level#	AUC
1	0.88±0.10	22	0.93±0.06
2	0.90±0.05	23	0.89±0.05
3	0.90±0.06	24	0.92±0.05
4	0.91±0.06	25	0.93±0.04
5	0.95±0.04	26	0.92±0.04
6	0.93±0.04	27	0.92±0.03
7	0.93±0.07	28	0.93±0.04
8	0.93±0.06	29	0.93±0.04
9	0.94±0.03	30	0.93±0.03
10	0.90±0.07	31	0.93±0.03
11	0.92±0.06	32	0.94±0.06
12	0.90±0.05	33	0.90±0.05
13	0.93±0.05	34	0.93±0.03
14	0.94±0.04	35	0.94±0.04
15	0.92±0.04	36	0.93±0.04
16	0.93±0.03	**37**	**0.95±0.03**
17	0.94±0.03	38	0.92±0.06
18	0.90±0.06	39	0.90±0.04
19	0.90±0.05	40	0.92±0.03
20	0.93±0.04	41	0.93±0.04
21	0.92±0.01	42	0.90±0.04

Table 13.3: *AUC for SVM presented with graph-based features and interest-based numerical features refined using books ontology "O1"*

Level#	AUC	Level#	AUC
1	0.93±0.04	8	0.92±0.04
2	0.91±0.04	9	0.89±0.07
3	0.92±0.04	10	0.84±0.07
4	0.94±0.04	11	0.93±0.03
5	0.89±0.07	12	0.89±0.08
6	0.91±0.04	**13**	**0.96±0.02**
7	0.90±0.04	14	0.93±0.04

Table 13.4: *AUC for SVM presented with graph-based features and interest-based numerical features refined using books ontology "O2"*

Level#	AUC	Level#	AUC
1	0.91±0.06	15	0.91±0.04
2	0.90±0.07	16	0.94±0.03
3	0.92±0.03	17	0.92±0.04
4	0.91±0.04	18	0.91±0.05
5	0.92±0.05	19	0.92±0.03
6	0.91±0.06	20	0.91±0.05
7	0.92±0.05	21	0.91±0.05
8	**0.95±0.02**	22	0.92±0.03
9	0.92±0.04	23	0.93±0.05
10	0.90±0.05	24	0.92±0.03
11	0.92±0.04	25	0.92±0.05
12	0.94±0.03	26	0.93±0.04
13	0.92±0.04	27	0.91±0.04
14	0.94±0.04	28	0.92±0.03

Table 13.5: *AUC for SVM presented with graph-based features and interest-based numerical features refined using movies ontology "O1"*

Level#	AUC	Level#	AUC
1	0.92±0.05	8	0.92±0.04
2	0.90±0.04	9	0.94±0.03
3	0.93±0.04	10	0.94±0.04
4	0.91±0.05	11	0.92±0.06
5	0.93±0.02	12	0.91±0.06
6	0.90±0.04	**13**	**0.95±0.03**
7	0.91±0.05		

Table 13.6: *AUC for SVM presented with graph-based features and interest-based numerical features refined using movies ontology "O2"*

Level#	AUC	Level#	AUC
1	0.94±0.02	22	0.91±0.04
2	0.93±0.02	23	0.92±0.06
3	0.91±0.06	24	0.91±0.05
4	0.92±0.04	25	0.91±0.04
5	0.94±0.05	26	0.95±0.03
6	0.91±0.05	27	0.92±0.03
7	0.91±0.05	28	0.92±0.04
8	0.91±0.10	29	0.91±0.04
9	0.91±0.06	30	0.90±0.07
10	0.88±0.06	31	0.94±0.04
11	0.94±0.03	32	0.94±0.03
12	0.93±0.05	33	0.92±0.04
13	0.91±0.04	34	0.90±0.03
14	0.89±0.06	35	0.93±0.04
15	0.92±0.04	36	0.89±0.09
16	0.93±0.04	37	0.93±0.03
17	0.89±0.06	38	0.91±0.04
18	0.92±0.04	**39**	**0.96±0.02**
19	0.91±0.07	40	0.91±0.04
20	0.93±0.04	41	0.94±0.03
21	0.90±0.03		

Table 13.7: *AUC for SVM presented with graph-based features and interest-based numerical features refined using words ontology "O1"*

Level#	AUC	Level#	AUC
1	0.93±0.03	9	0.92±0.04
2	0.88±0.03	10	0.91±0.04
3	0.94±0.03	11	0.89±0.06
4	0.92±0.06	12	0.89±0.07
5	**0.95±0.04**	13	0.88±0.04
6	0.92±0.06	14	0.92±0.04
7	0.93±0.03	15	0.92±0.03
8	0.92±0.03		

Table 13.8: *AUC for SVM presented with graph-based features and interest-based numerical features refined using words ontology "O2"*

Level#	AUC	Level#	AUC
1	0.93±0.03	16	0.91±0.04
2	0.92±0.03	17	0.93±0.05
3	0.90±0.05	18	0.95±0.02
4	0.93±0.04	19	0.92±0.04
5	0.89±0.04	20	0.92±0.04
6	0.91±0.05	21	0.91±0.07
7	0.91±0.05	22	0.93±0.03
8	0.93±0.03	23	0.91±0.06
9	0.92±0.06	24	0.91±0.06
10	0.90±0.06	25	0.91±0.04
11	0.90±0.06	26	0.92±0.03
12	**0.95±0.02**	27	0.94±0.03
13	0.93±0.05	28	0.94±0.03
14	0.92±0.04	29	0.93±0.03
15	0.91±0.03		

Table 13.9: *AUC for SVM presented with graph-based features and interest-based numerical features refined using phrases ontology "O1"*

Level#	AUC	Level#	AUC
1	0.93±0.03	7	0.92±0.04
2	0.92±0.05	8	0.91±0.03
3	0.92±0.05	9	0.92±0.04
4	0.92±0.04	10	0.92±0.03
5	0.92±0.06	**11**	**0.93±0.03**
6	0.93±0.04	12	0.91±0.04

Table 13.10: *AUC for SVM presented with graph-based features and interest-based numerical features refined using phrases ontology "O2"*

Level#	AUC	Level#	AUC
1	0.91±0.05	10	0.91±0.04
2	0.90±0.06	11	0.94±0.03
3	0.93±0.05	12	0.93±0.04
4	0.90±0.02	13	0.94±0.03
5	0.91±0.05	14	0.90±0.05
6	0.93±0.04	15	0.94±0.04
7	0.93±0.04	16	0.90±0.06
8	0.94±0.03	17	0.91±0.03
9	0.92±0.03	**18**	**0.95±0.03**

Chapter 14

Appendix E

In this Chapter, we report the AUC values of various classifiers at the task of predicting interests of users of the *Live Journal* social network (see Sections 5.2 and 6.2). The task of predicting interests is performed for two sets of interests. "Set 1" comprises of four interests namely *Books*, *Movies*, *Words* and *Phrases*. And "Set 2" comprises of nineteen interests namely *Movies*, *Words*, *Phrases* and sixteen genres into which *Books* are divided. Tables 14.1, 14.2 and 14.3 list the AUC values for classifiers at the task of predicting interests from "Set 1" when presented with interest-based nominal features, interest-based numerical features and a combination of the two respectively.

Table 14.1: *AUC for different classifiers presented with interest-based nominal features*

Interest	SVM	Logistic	J48	RandomForest	OneR
Books	0.486	0.532	0.486	0.505	0.5
Movies	0.493	0.499	0.49	0.47	0.498
Words	0.502	0.497	0.492	0.474	0.5
Phrases	0.616	0.611	0.522	0.56	0.526
Average	0.524	0.5348	0.498	0.50225	0.506

In addition, Tables 14.4, 14.5 and 14.6 show the AUC values for classifiers at the task of predicting interests from "Set 1" when presented with interest-based nominal features, interest-based numerical features and a combination of the two respectively.

Table 14.2: *AUC for different classifiers presented with interest-based numerical features*

Interest	SVM	Logistic	J48	RandomForest	OneR
Books	1	0.988	0.99	0.995	0.995
Movies	0.997	0.989	0.982	0.994	0.978
Words	0.993	0.981	0.942	0.981	0.962
Phrases	0.992	0.991	0.988	0.995	0.982
Average	0.996	0.9873	0.976	0.99125	0.97925

Table 14.3: *AUC for different classifiers presented with interest-based nominal and numerical features*

Interest	SVM	Logistic	J48	RandomForest	OneR
Books	1	0.993	0.99	0.995	0.995
Movies	0.997	0.993	0.982	0.994	0.978
Words	0.993	0.954	0.95	0.981	0.962
Phrases	0.992	0.991	0.988	0.995	0.982
Average	0.996	0.9828	0.978	0.99125	0.97925

Table 14.4: *AUC for different classifiers presented with interest-based nominal features*

Interest	SVM	Logistic	J48	RandomForest	OneR
Books1	0.532	0.64	0.618	0.599	0.498
Books2	0.594	0.556	0.497	0.552	0.494
Books3	0.526	0.6	0.559	0.575	0.517
Books4	0.575	0.598	0.635	0.633	0.496
Books5	0.523	0.526	0.526	0.552	0.495
Books6	0.606	0.6	0.535	0.57	0.543
Books7	0.551	0.556	0.519	0.596	0.531
Books8	0.574	0.564	0.497	0.536	0.469
Books9	0.62	0.618	0.554	0.562	0.539
Books10	0.544	0.539	0.557	0.55	0.546
Books11	0.528	0.536	0.508	0.484	0.54
Books12	0.594	0.583	0.543	0.574	0.513
Books13	0.544	0.544	0.502	0.598	0.503
Books14	0.574	0.582	0.484	0.575	0.514
Books15	0.543	0.53	0.507	0.554	0.525
Books16	0.554	0.547	0.503	0.597	0.498
Movies	0.532	0.553	0.478	0.588	0.496
Words	0.487	0.535	0.477	0.574	0.498
Phrases	0.634	0.616	0.556	0.59	0.527
Average	0.56	0.5696	0.529	0.57153	0.512737

Table 14.5: *AUC for different classifiers presented with interest-based numerical features*

Interest	SVM	Logistic	J48	RandomForest	OneR
Books1	0.843	0.877	0.779	0.8	0.699
Books2	0.761	0.853	0.735	0.747	0.633
Books3	0.877	0.879	0.793	0.813	0.734
Books4	0.865	0.863	0.74	0.789	0.721
Books5	0.866	0.865	0.766	0.792	0.757
Books6	0.88	0.877	0.79	0.842	0.769
Books7	0.908	0.903	0.823	0.87	0.783
Books8	0.916	0.914	0.838	0.866	0.811
Books9	0.926	0.925	0.861	0.887	0.83
Books10	0.938	0.938	0.843	0.897	0.838
Books11	0.937	0.936	0.857	0.913	0.842
Books12	0.929	0.928	0.835	0.879	0.817
Books13	0.959	0.951	0.932	0.916	0.861
Books14	0.969	0.97	0.917	0.911	0.835
Books15	0.983	0.987	0.942	0.963	0.857
Books16	0.984	0.987	0.916	0.971	0.889
Movies	0.996	0.997	0.936	0.975	0.912
Words	0.995	0.993	0.929	0.979	0.921
Phrases	0.904	0.911	0.816	0.865	0.815
Average	0.918	0.9239	0.845	0.87763	0.806526

Table 14.6: *AUC for different classifiers presented with interest-based nominal and numerical features*

Interest	SVM	Logistic	J48	RandomForest	OneR
Books1	0.875	0.875	0.776	0.816	0.699
Books2	0.783	0.851	0.718	0.805	0.633
Books3	0.887	0.889	0.784	0.863	0.734
Books4	0.867	0.861	0.781	0.85	0.721
Books5	0.868	0.856	0.775	0.808	0.757
Books6	0.883	0.891	0.765	0.863	0.769
Books7	0.908	0.9	0.808	0.875	0.783
Books8	0.916	0.917	0.826	0.875	0.811
Books9	0.926	0.921	0.855	0.892	0.83
Books10	0.938	0.934	0.843	0.91	0.838
Books11	0.937	0.925	0.835	0.91	0.842
Books12	0.929	0.919	0.865	0.891	0.817
Books13	0.958	0.948	0.873	0.934	0.861
Books14	0.967	0.971	0.902	0.927	0.835
Books15	0.983	0.982	0.916	0.957	0.857
Books16	0.985	0.97	0.916	0.977	0.899
Movies	0.996	0.968	0.944	0.984	0.912
Words	0.995	0.964	0.903	0.97	0.921
Phrases	0.905	0.915	0.837	0.88	0.815
Average	0.921	0.9188	0.838	0.89405	0.807053

Chapter 15

Appendix F: Visualization of the Ontology

Visual inspection of an ontology can provide useful insights about how various instances have been organized into concepts that capture the semantic knowledge in a domain. We have used the open source tool "Cytoscape" [Shannon et al., 2003] to inspect the ontology of interests constructed by our algorithm. Because HAD involves minimal human intervention during the ontology building process (Divisive Clustering step, see Section 3.2), semantically incorrect concepts may get introduced into the hierarchy. In principle, visual inspection of the ontology is useful in finding such incorrect concepts and tools like Cytoscape can help editing and correcting the ontology by adding/deleting nodes and edges. Cytoscape scales to very large networks (our Ontology consists of about $85,000$ nodes), and by using Cytoscape, the ontology and its associated annotations can be imported from or exported to standard file formats such as Graph-ML (GML), Simple Interaction Format (SIF), and even to simpler file types like delimited text files and excel workbooks. Figure 15.1 shows a screen-shot of this tool.

We have not edited the ontology returned by HAD, but have performed visual analysis to get an idea about its quality.

One small subset of the ontology constructed from the *Live Journal* data using HAD is shown in Figure 15.2, which delineates the organization of several interests related to the

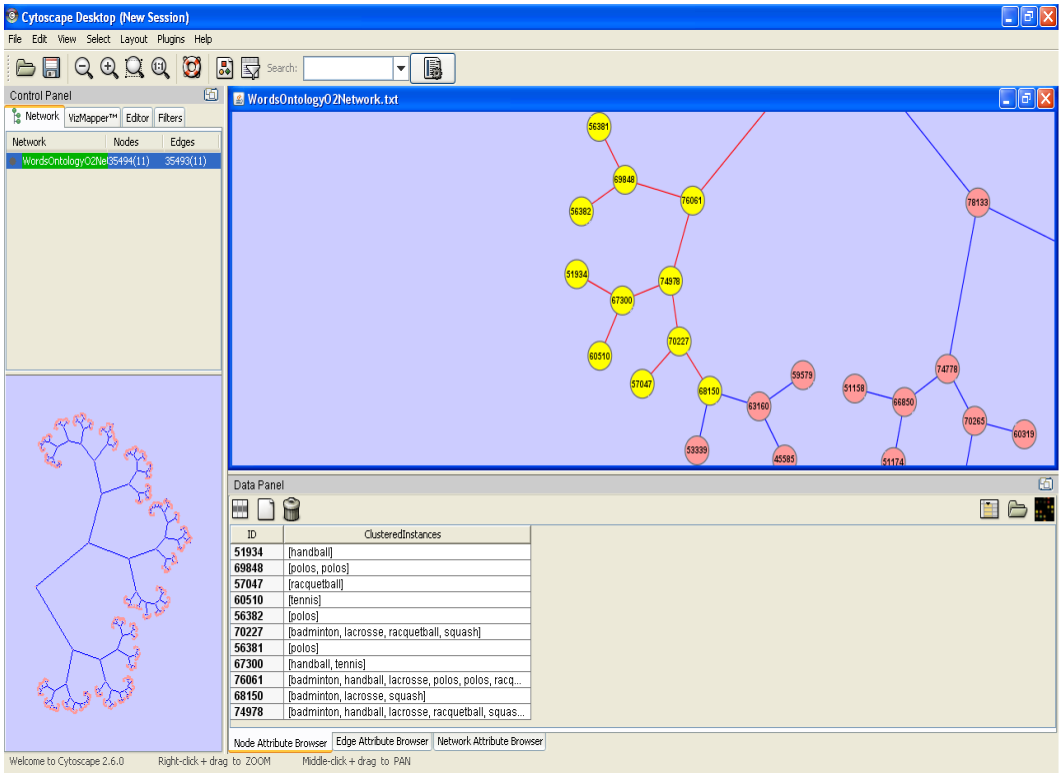

Figure 15.1: *Cytoscape open source tool-kit*

concept of "networking". It is comprised of the following concepts:

```
1: networks, topology, networks, networking, wireless,
radio, networks, wifi
2: networks, networks, networking, topology
3: radio, networks, wireless, wifi
4: networks, topology
5: networks, networking
6: radio, wireless
7: networks, wifi
8: networks (defined as)-
net| network| mesh| meshing| meshwork| wire
9: topology (defined as)-
```

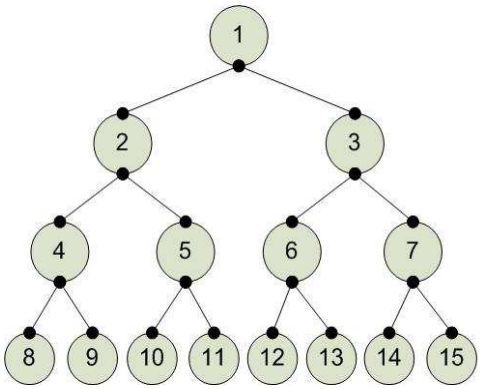

Figure 15.2: *Ontology of terms related to "networking"*

```
network| components| connected| mesh| wire
10: networks (defined as)-
network| communicate| group
11: networking (defined as)-
network| communicate| group
12: wireless
wireless| communication| electromagnetic| waves
13: radio (defined as)-
wireless| communication| based| broadcasting|
electromagnetic|waves| transmit| radio| waves
14: networks (defined as)-
broadcasting| communication| stations|
transmit| programs
15: wifi (defined as)-
wireless| network| high| frequency| radio|
signals|transmit| receive| data| broadcasting
```

Figure 15.2 shows that there are several instances describing the "networking" interests and they combine together (based on similarity) in a meaningful way to form more abstract concepts.