Learning Models for Multi-Source Integration*

Sheila Tejada Craig A. Knoblock Steven Minton
University of Southern California/ISI
4676 Admiralty Way
Marina del Rey, California 90292
{tejada,knoblock,minton}@isi.edu

Abstract

One issue involved in accessing multiple heterogeneous information sources is how to integrate the retrieved data. SIMS, an information mediator, handles this problem by mapping the data in each information source into a common information or domain model. This model defines the relationships between the data in the different sources, so that the data can be integrated properly. Human experts who are familiar with the contents of the information sources are required to generate the model and perform the mapping of the sources. Automating this process of constructing the model of the information sources would be more convenient and efficient. Assuming that the structure of the information sources is a set of tables, the first step in this process is mining the tables to discover relationships in the data. These relationships are then used to further develop the model by helping to determine the necessary classes, superclass/subclass relationships, and associations.

Introduction

Because of the growing number of information sources available through the internet there are now many cases in which information needed to solve a problem or answer a question is spread across several information sources. In order to obtain the desired information, multiple sources need to be accessed, and the retrieved data then has to be integrated. An example to illustrate this problem of multi-source integration is when given two information sources, one source containing information about comic books and the other about super heroes, and you want to ask the question "Does Spiderman appear in a Marvel comic book?" To answer this question both of the information sources need to be accessed. In this situation it is necessary to have information about the relationships of the data within each source, as well as between sources, to properly access and integrate the data retrieved. The SIMS (Ambite *et. al.* 1995) information broker captures this type of information in the form of a model, called a domain model. Because all the information sources map into the domain model, the domain model serves as an interface which can provide the user access to multiple sources. The user specifies transactions to be performed using the terms of the domain model, and is not required to have knowledge about specific sources.

In the next section we will describe the domain model in more detail, and then discuss our basic approach to address this problem of learning models for multi-source integration. Next, we will present our preliminary work that we have conducted on creating abstract descriptions of the data in order to mine for existing relationships. We will then provide a discussion of related work, and conclude by describing the future directions of our work.

*This research reported here was suppoerted in part by Rome Laboratory of the Air Force Systems Command and the Advanced Research Projects Agency under Contract Number F30602-94-C-0210, and in part by an Augmentation Award for Science and Engineering Research Training funded by the Advanced Research Projects Agency under Contract Number F49620-93-1-0594. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing the officail opinion or policy of RL, ARPA, the U.S. Governments, or any person or agency connected with them.

Model Description

The diagram shown in Figure 1 is a partial model of both the Comic Book and Super Hero information sources.

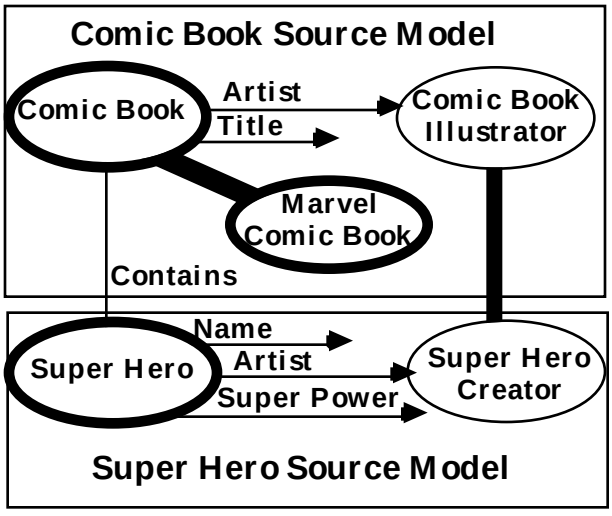

Figure 1

This model defines the relationships of the data within the two sources and also shows their interaction. The ovals in the diagram represent classes and the lines describe the relationships between the classes. There are two types of classes -- a basic class and a composite class. Members of a basic class are single-valued elements, such as strings or numbers; while the members of a composite class are objects which can have several attributes.

In this example **Comic Book Illustrator** and **Super Hero Creator** are both basic classes which contain artists' names. **Comic Book**, **Marvel Comic Book**, and **Super Hero** are all represented as composite classes. The arrows represent the attributes of the objects, e.g. **Artist**, **Title**, **Name**. The **Artist** arrow between **Super Hero** and **Super Hero Creator** means that the values of that attribute belong to the set of names of **Super Hero Creator.** The thick lines in the model represent superclass/subclass relationships,e.g the names of **Super Hero Creator** are a subset of **Comic Book Illustrator**. The **Contains** relationship between **Comic Book** and **Super Hero** is an association link, which links a **Comic Book** object with a **Super Hero** object. This is like an attribute link, except that it connects two composite classes.

Learning the Model

Presently, domain models are manually generated by human experts who are familiar with the data stored in the sources. Automation of this task would improve efficiency and the quality of the model. We have conducted preliminary work in automating the process of model construction. The basic idea for our approach is that we examine the data in the information sources to extract the relationships needed to create the domain model.

Learning the model is an iterative process which involves user interaction. The user can browse the generated model to make corrections or specify an information source to be added or deleted. The model proposed to the user is generated by heuristics we have developed to derive the necessary relationships from the data. The user's input is incorporated in generating this model. The heuristics that we have applied involve creating abstract descriptions of the data which is described further in the next section.

Abstract Descriptions

We have assumed that the data is stored as tables, and that values within a column of a table are related to each other. An abstract description of a column of data defines the basic class for that collection of data. Properties or features of the data, such as type and range, are contained in the abstract descriptions. These descriptions are then used to help determine the superclass/ subclass relationships that exist between the basic classes. The descriptions can constrain the number of possible related classes. Potentially, all classes would need to be checked for a superclass/subclass relationship, but now only those classes that have similar descriptions are tested. One of the main reasons to generate these abstract descriptions of the data are so that they can be used to help determine composite classes and their relationships.

Generating Abstract Descriptions

An example of a data table is shown in Figure 2. A abstract description is determined for each column or attribute of the table. In this table the set of attributes are **Name, Artist,** and **Super Power**. A set of features is computed for each of these attributes which characterizes the basic classes to which each attribute belongs. The set of features that describes the basic classes also helps in discovering the relationships between basic classes. The relationships between basic classes should reflect the relationships that exist between the attributes.

Name	Artist	Super Power
Spiderman	Stan Lee	Spider Sense
Spawn	Todd McFarlane	Immortality
Cyclops	Stan Lee	X-ray Power
.	.	.
.	.	.

Super Hero Table

Figure 2

Shown in Figure 3 is the table that contains the computed set of features for the attributes given in Figure 2. The set of features used to characterize each attribute in this example is Unique, Type, Subtype, Length, and Set of Values.

	Unique	Type	Subtype	Length	Set of Values
Name	Yes	String	Alpha-betic	6<=L<=9	{Spawn, . . .}
Artist	No	String	Alpha-betic	8<=L<=14	{Stan Lee, . . .}
Super Power	Yes	String	Alpha, Other	11<=L<=12	{Spider Sense.. .}

Figure 3

The Unique feature determines whether there are duplicate elements, e.g. the **Name** attribute does not contain duplicate values. The Type of an attribute is either **string** or **number**. The values for the feature Subtype are subtypes of **string** and **number**. The subtypes for **number** are **integer** and **real**. For **string** there are four basic subtypes: **alphabetic, alphanumeric, numeric**, and **other**. The value **other** refers to a string that has a least one character which is not a letter, a space or a number. The collection of the values of an attribute may be of one subtype or any combination of the four, as shown by the **Super Power** attribute.

The Length or Range feature determines the range of the length of the set of strings or the range of the set of numbers in the attribute. For each of the attributes a set of their values are calculated by the Set of Values feature; therefore, duplicate values are removed, as is done for the Artist attribute. There are many other features which can be used to find regularities in the data, such as determining if the alphabetic strings of an attribute are words. These features represent only a subset.

Relationships between Basic Classes

The set of features shown in Figure 3 are the abstract descriptions of the attributes. These descriptions define the basic class of the attribute. The Set of Values feature provides all of the instances belonging to that basic class. The Type and Length features can be used to restrict the set of attributes that are related to a given attribute. For the attribute **Name**, the value for Type is **string**, and the value for Subtype is **Alphabetic**; therefore, **Name** can only be related to other attributes whose basic classes also have Type as **string** and a Subtype which includes **Alphabetic**.

Based on the table in Figure 3 the set of related attributes whose basic classes overlap with **Name**'s Type feature is {**Artist, Super Power**}. The same is now done for the Length feature of **Name**. There are no basic classes whose Length feature contains or is contained in the **Name**'s Length feature, as shown by the comparisons of Length features depicted in Figure 4; therefore, the set of related attributes is empty.

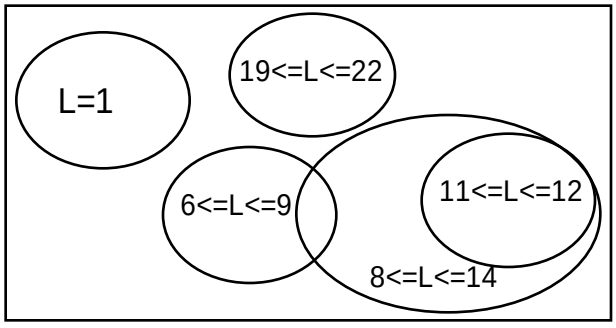

Figure 4

Next, the Set of Values feature is evaluated using the set of attributes produced by the intersection of the two previous sets of related attributes. This feature will decide the type of relationship that exists between the basic classes. The possible types of relationships are superclass/subclass, overlapping classes, and non-overlapping related classes. The Set of Values feature of **Name** is evaluated by testing the overlap of its values with those of an attribute in the related set.

When the set of values of an attribute is a subset (or superset) of another attribute, then a superclass/subclass relationship exists between their basic classes. An example of this type of relationship can be seen in Figure 1 between the basic class **Super Hero Creator** of the **Super Hero**.**Artis**t attribute and the **Comic Book Illustrator** class of the **Comic Book**.**Artist.** The attributes are mapped into the model using this process which is repeated for each attribute in the table. For the attributes that have an overlapping or non-overlapping class relationship, a superclass of the two attributes needs to be created to properly represent their interaction. Further information needs to be computed in order to show that the set of these types of proposed relationships are valid.

Related Work

There has been similar work conducted in this area (Perkowitz & Etzioni 1995), but it has focused on learning only the attributes of the information sources. Their approach also assumes that it has instances as well as an initial model of the information to be learned from perspective information sources. Related work using mining techniques to learn a model of the information sources has been researched by Kero et al (Kero *et al*. 1995). Their work uses data mining to determine which attributes are related, while our work described in this paper is primarily interested in the types of the relationships that exist between the attributes.

Discussion

This is preliminary work on the automation of modeling information sources, which has focused mainly on discovering the relationships between basic classes. Further work still needs to be conducted to devise mechanisms for deciding when to generate a primitive superclass for the overlapping and the non-overlapping related basic classes. We are planning to apply statistical methods to assist in addressing these cases, as well as integrating a natural language knowledge base into this process to help determine whether classes are semantically related. The relationships between the basic classes will be useful in the later steps of the modeling process which are to determine the associations and superclass/subclass relationships between composite class.

References

Kero, B., Russell, L., Tsur, S., & Shen W. An Overview of Database Mining Techniques. The KDD workshop in the 4th International Conference on Deductive and Object-Oriented Databases, Singapore, 1995.

Ambite, J., Y. Arens, N. Ashish, C. Chee, C. Hsu, C. Knoblock, and S. Tejada. The SIMS Manual, Technical Report ISI/TM-95-428,1995.

Perkowitz, M. & Etzioni, O. Category Translation: Learning to Understand Information on the Internet. The International Joint Conference on Artificial Intelligence, Montreal, 1995.