Generating Association Rules from Semi-Structured Documents Using an Extended Concept Hierarchy

Lisa Singh
Peter Scheuermann
Bin Chen

Department of Electrical and Computer Engineering
Northwestern University
Evanston, IL 60208
{lsingh, peters, bchen}@ece.nwu.edu

Abstract

Most data mining research has focused on generating rules within databases containing structured values while essentially ignoring the potentially valuable information that exists in the unstructured blocks of text. This paper suggests an approach for generating association rules that relates structured data values to concepts extracted from unstructured data. Our approach involves the use of an extended concept hierarchy (ECH) to maintain parent, child, and sibling relationships between concepts. This structure allows us to generate rules that relate a given concept in the ECH and a given structured attribute value to the neighbors of the given concept in the ECH. We also describe an efficient implementation of the ECH that keeps track of concepts and pointers to documents associated with them. Experimental results on documents from the ABI/Inform Information Retrieval System are presented.

1 Introduction

With the abundant amounts of information available to businesses today, an urgent need exists to develop tools that extract knowledge from large data sources, including on-line databases, data warehouses, and the Internet [PBKK96]. Potentially, businesses may have hundreds or thousands of data sources, each organizing data to best support individual day to day functions. Much data exists in well structured databases, but large amounts still reside in ill-structured legacy systems or partially structured textual document systems. This information is potentially an invaluable source for analysis and decision support. Distributions of attribute values within/across data sources or associations between different structured and unstructured data components can be used to evaluate trends, predict markets, classify customer groups, and develop generalizations. In general, this type of analysis has been coined "knowledge discovery in databases". Knowledge discovery in databases (KDD) is the process of extracting higher level knowledge by identifying hidden patterns within large data sets [FSS96].

The KDD process involves a number of steps including data cleansing, data reduction, data transformation, data mining and data interpretation [FSS96]. This paper will focus on the data transformation and data mining phases in the context of semi-structured data. During the data transformation phase, we manually create an extended concept hierarchy (ECH) by extracting concepts from the unstructured components of documents and associating these concepts based on semantic relationships between them. At the conclusion of the data transformation step, each concept in the ECH maintains pointers to related concepts and pointers to documents containing the concepts. In the data mining phase, we discover qualitative and quantitative associations between concepts in the ECH and values of structured attributes in the database. We accomplish this by using set operators to compare documents associated with different concepts in the ECH to documents associated with values of attributes in the database. Because the ECH stores relationships among concepts, we can generate rules that cannot be discovered using the database alone.

The remainder of this paper is organized as follow. Section 2 contains a motivating example which identifies meaningful rules involving concepts extracted from text. Section 3 defines relevant background concepts. In Section 4, related literature is reviewed. In Section 5, a method for constructing the ECH and algorithms for generating a set of qualitative and quantitative association rules are presented. In Section 6, we present a performance study that investigates different size ECHs, different number of relationships among concepts, and different types of association rules. Section 7 concludes by discussing extensions of this work for the World Wide Web (WWW) and digital library systems.

2 Motivating Example

Magazine articles, research papers, and World Wide Web HTML pages are traditionally considered semi-structured information. Each of these examples contains some clearly identifiable features, including author, date, and publisher or WWW address. They also include blocks of text that are considered unstructured components of the documents.[1] Although these documents are not well structured, we claim that they contain useful information.

The problem with document data is that limited insight about a document can be attained using only the structured document components. In a digital library system, a user can find all documents written by a particular author, involving a keyword and/or appearing in a particular periodical. All of these tasks help users identify articles meeting a certain criteria. However, these operations only provide limited knowledge about the document collection as a whole. Generating rules about a document collection or a subset of the collection can help users answer the following questions:

- Which journal typically publishes articles associated with my research?
- Which authors publish most frequency in my area of expertise?
- Which journals typically maintain a broad range of topics?
- How are the articles in my research area broken down?

If structured document concepts are maintained in a traditional database, some quantitative rules can be generated using basic SQL operations. For example, we can determine the percentage of articles published in a particular research area, or the percentage of articles published by a particular author in a specific research area:

Rule A*: 20% of the articles written by Joe Smith involve C++ programming.*

[1] Note, graphics and audio files are also considered unstructured information, but this paper focuses on textual blocks. The mining techniques can be extended for these other data types.

Joe Smith → C++ programming : 20%

However, without knowledge of relationships that exist among concepts, we cannot generate rules relating multiple concepts to an attribute value. As an example, given a document collection containing business abstracts and journal articles, potential article submitters or journal editors might be interested in the following association rule:

Rule B: *45% of the **corporate profile** articles published in **Harvard Business Review** focus on **reengineering**, while only 5% discuss **human resource management**.*

Harvard Business Review Λ *corporate profile →*
reengineering : 45%
human resource management : 5%

Corporate profile, *reengineering*, and *human resource management* are all concepts representing unstructured blocks of text. In contrast, *Harvard Business Review* may be stored as a structured value in the database, i.e. publication. Rule B assumes the following knowledge:

- Corporate Profile relates to Reengineering
- Corporate Profile relates to Human Resource Management
- Corporate Profile is broader than Reengineering
- Corporate Profile is broader than Human Resource Management

Without knowledge about relationships among concepts, we are unaware of which combination of concepts generate the most interesting rules. This example illustrates the potential to generate valuable rules about a document collection given a structure that maintains concepts and relationships among them. The structure we choose to maintain this information is an extended concept hierarchy.

3 Background Concepts

3.1 Extended Concept Hierarchy: Motivation & Overview

Background knowledge, additional information that is domain specific, is typically provided by a domain expert. Although not necessary, it enables a data mining tool to extract non-primitive rules from the database. One representation of background knowledge is a concept hierarchy. Concept hierarchies define a sequence of mappings from low level concepts (i.e. data values in the database) to their high level counterparts [HF94]. Figure 1 shows an example of a simple concept hierarchy for an employee database.

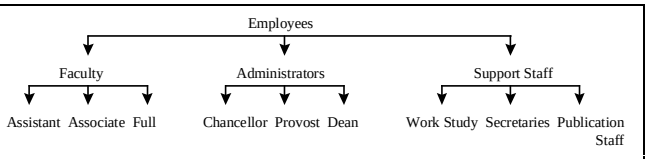

Figure 1 : University Employee Concept Hierarchy

The concept hierarchy in Figure 1 is a tree structure. However, a concept hierarchy is not constrained to a tree structure. It can be as simple as a linked list, and as complex as a lattice or arbitrary graph structure. Our data set does not guarantee a unique root. However, in an effort to generate more interesting rules, we model parent, child and sibling relationship. Figure 2 illustrates an example of this. Therefore, we refer to our concept structure as an extended concept hierarchy (ECH). Figure 3 shows an example from a business document collection that contains parent, child and sibling relationships.

If we assume that data values exist only at the leaf nodes of the concept hierarchy in Figure 1, the employee records would contain an attribute that specifies an employee's job title. Figure 4 illustrates this. The remaining values in the tree are not stored in the database and therefore, are not explicit data values. Instead, these values are considered domain knowledge identified by an expert.

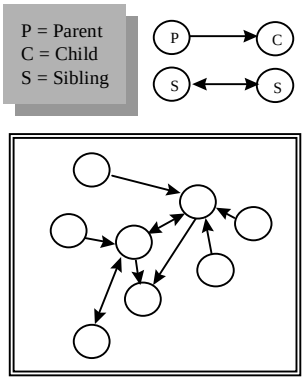

Figure 2 : Concept Structures

Although data values typically reside only at the leaf nodes, data values can exist at any level of the concept hierarchy. If we assume that all levels of the concept hierarchy represent data values, then the employee table may resemble that shown in Figure 5. Notice that in this approach, attribute values exist at each level of the concept hierarchy. For this particular example, nonleaf node values are duplicated for each employee record. To avoid this redundancy, this information can be placed in a lookup table. However, if a child concept, i.e. secretaries, has two or more parent concepts, i.e. administrators and support staff, then we must maintain an attribute that identifies the correct parent for each record. Since we are generating an ECH to represent the unstructured component of the data set each node will contain a concept.

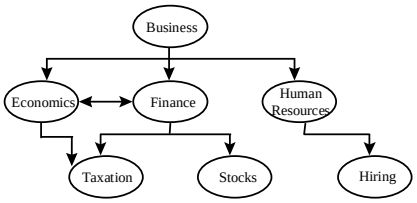

Figure 3 : Data Set Example of ECH

Consequently, our structure is mapping data values to every node. Associated with each concept in Figure 3 is a list of documents containing the concept. Section 5.2 discusses the construction of the ECH in more detail.

The next section describes how a concept hierarchy or an ECH can aid in the knowledge discovery process by finding higher level associations that are not apparent when viewing raw data values. Integrating background knowledge with data mining algorithms generates more interesting, nontrivial patterns.

3.2 Association Rules: Definition & Examples

[AIS93] first introduced mining for association rules in the context of a grocery store transaction database. Association rules identify groupings between sets of items with some *minimum specified confidence*, where *confidence* is defined as the percentage of objects satisfying a rule [HF95]. Only sets of items satisfying a *minimum support* condition are potential rule candidates. [AIS93] define *support* as the percentage of group A in pattern space X, where X is the set of all patterns in the database for a particular attribute and A is a subset of those patterns. The following are examples of traditional association rules:

A. When AT&T stock rises a half a point and IBM stock drops a quarter of a point, Microsoft stock rises 60% of the time.

B. 80% of the customers renting Star Wars will also rent Empire Strikes Back and Return of the Jedi.

Given a database containing daily stock prices, Rule A searches for stock prices that regularly change together. The set of items in this case is AT&T, IBM, Microsoft. Assuming this set of items satisfies the minimum support condition, the data mining tool then determines the antecedent (AT&T, IBM) and the resultant (Microsoft) portions of the rule that generate a high enough confidence. Similarly, Rule B finds relationships among movies rented by customers, where Star Wars, Empire Strikes Back and Return of the Jedi comprise the item set. In both cases, items being related are values of a single attribute. The association rules generated in this paper are not of this traditional nature. Instead, they parallel association rules that incorporate values across multiple attributes. For example,

C. 80% of flower shops in Chicago sell geraniums and lilies.

If we assume that store type, location and product type are attributes within different tables in a relational database, then obtaining Rule C would require multiple joins. Perhaps a user specifies an interest in rules involving Chicago and geraniums. A data mining tool finds that geraniums and lilies are sold together with some minimum support. This in turn leads to an association rule with an 80% confidence:

flower shops Λ Chicago $\rightarrow$ geraniums and lilies

Depending upon the structure of the database, associations may or may not be easy for a data mining tool to identify. There are a few problems that arise when generating rules. First, the size of most corporate databases fall into the terabyte range. Therefore, an exhaustive search can consume a large number of resources. Data mining tools must be sophisticated enough to find interesting patterns quickly by decreasing the size of the pattern space. Another problem is that the data may not be structured in a manner that allows an analyst to extract a valuable pattern. Consequently, the data may have to be processed prior to any rule generation.

As an example, suppose that Rule C needs to be extracted from documents or WWW pages. Special processing would need to be done to determine which web sites correspond to flower shops. The next stage involves text extraction techniques that identify important concepts at each of these web sites, including types of flowers and location of flower shops. Only then can we generate a quantitative rule that determines the percentage of flower shops in Chicago selling geraniums and lilies. This approach is unrealistic in an online environment. Once a user submits a request, too much time would be expended extracting concepts for the documents or WWW pages. For this reason, preprocessing of semi-structured data is necessary. Concept extraction and concept relationship generation should be part of an off-line process. Therefore, the generation of our ECH is a preprocessing step, completed prior to the data mining phase of the KDD process.

Similar to Rule C, we generate rules that relate a structured attribute with concepts in the ECH. A user provides a starting concept, a structured component, and a minimum confidence. Our tool then generates rules above the user specified minimum confidence using the concept, its neighbors in the ECH that have a relationship above a system defined minimum support, and the structured attribute values. Section 5.2 includes a more detailed example. For our database, we define *support* between two concepts to be the ratio between the documents in which both

Employee Number	Position
21001	Associate
75675	Assistant
32654	Dean
67381	Work Study
98670	Full Prof

Figure 4 : Employee Table Mapping to Leaf Node of Concept Hierarchy

Employee Number	Position	Staff Category
21001	Associate	Faculty
75675	Assistant	Faculty
32654	Dean	Admin
67381	Work Study	Support Staff
98670	Full Prof	Faculty

Figure 5 : Employee Table Mappings to Entire Concept Hierarchy

concepts, C_a and C_b , occur in to the set of documents both concepts appear in:

$$\frac{\text{documents } (C_a) \cap \text{documents } (C_b)}{\text{documents } (C_a) \cup \text{documents } (C_b)}$$

Because of the structure of most document collections, association rules involving multiple concepts and structured document attributes cannot be determined using standard SQL operations. Instead, it is necessary to identify meaningful concepts associated with each document and determine significant relationships among concepts. A significant relationship corresponds to a support above the minimum specified. As previously mentioned, we store this type of information in an ECH. Once the ECH is created, attribute values of different attributes in the database can be associated with different subsets of concepts in the ECH.

4 Related Literature

Extensive research has been reported on algorithms that discover association rules [AIS93, AS94, HF94, HF95, PCY95, SA95, KMR94]. All of these papers focus on generating rules using a transaction database. Specifically, they investigate efficient methods for generating large item sets within a set of transactions, where a transaction corresponds to an attribute with repeating values. These item sets are then used as the basis for identifying patterns. We do not generate large item sets for two reasons. First, the user submits a starting attribute value and concept. Once the starting point has been identified, the ECH provides us with the necessary concept relationships. Also, our association is between a structured attribute and a set of concepts present in the ECH. Therefore, the rules we are attempting to generate are different than those described in [AIS93, AS94, HF94, HF95, PCY95, SA95, KMR94].

A few of these papers incorporate the use of a concept hierarchy [HF94, HF95, FL96]. However, in all of these cases, the concept hierarchy is in the form of a tree and data values only exist at the leaf node level. Further, it is typically used to generalize the association rule. [HF95] focus on generating association rules by generalizing data values to a particular level in the concept hierarchy. We use it to identify a set of concepts that can be used in the same rule, thereby generating a single rule that involves multiple levels of the ECH. Because we only investigate neighboring concepts of the given concept, overgeneralization is not a problem.

[TPL95, LHKK96, FD95] attempt to discover interesting rules using semi-structured data. [TPL95, LHKK96] propose

algorithms for classifying semi-structured data. Our paper differs since we are focusing on association as opposed to classification. [FD95] also use a concept hierarchy to describe the contents of articles. However, their goal is to study concept distributions within the document collection. In their model, each concept node is a discrete random variable whose values are identified by its children. Then probability distributions are created based on the proportion of documents identified by different concepts. Once the distribution is created, it is compared to other well defined distributions, including the uniform distribution. In this manner, the user can generate statistics about the document set. For example, given a concept *computer*, we could determine whether the distribution of the types of computers deviates from that of a normal distribution. Although the framework of [FD95] is similar to ours, our goal is to associate structured values to the concepts in the ECH, rather then generate statistics about the concepts themselves.

5 Approach & Algorithm

5.1 Description of Database & Data Set

Our data set resides in an Oracle 7 relational database on an HP 700 series workstation. The database consists of over 50,000 documents from the ABI/Inform Information Retrieval System. ABI/Inform maintains bibliographic information and abstracts of articles from over 800 journals. Our database contains only a small subset of the documents in ABI/Inform's IR system. ABI/Inform also includes a thesaurus, a hierarchical classification tree, and manually indexed subject headings. We use this additional information to manually construct an ECH consisting of approximately 250 concepts.

In our database, structured components of the documents are divided into multiple attributes, while unstructured components are placed as is into other fields. The publication, author, and location tables maintain structured data in the database. Title, abstract and document text are attributes that represent the unstructured data. Table 1 shows examples of structured and unstructured data. Both the title and abstract are maintained within the database, while the document itself is maintained outside the database. Only the document id is stored in the database.

STRUCTURED		***UNSTRUCTURED***	
Author	White, Michael.	**Title**	States' rights.
Publication	World Trade, 8(10): 33-34, 1995 Nov.	**Abstract**	According to Carol Conway of the Corporation for ...
Location	US	**Document**	Full Text

Table 1 : Structured & Unstructured Data

5.2 Construction of an Extended Concept Hierarchy

As previously mentioned, we construct an ECH to represent concepts that exist within different documents. Our ECH shows relationships between different concepts appearing in the unstructured components of the document collection. We identify three types of relationships: parent, child, sibling. Given a concept *p*, parents of *p* are defined to be related concepts that are semantically more general or broader than concept *p*. For example, if concept *p* is *subroutine*, then *program* may be viewed as a parent concept. Similarly, children concepts of *p* are related concepts that are narrower or more specific than *p*. If *p* is equal to *animal*, then *reptiles* and *mammals* would be considered children concepts. Finally, a sibling concept is defined to be a related concept that has a similar meaning to concept *p*. In some cases, it is a synonym. As an example, if *p* is *dog*, then *wolf* may be considered a sibling.

What constitutes a valid ECH in one database does not necessarily imply the validity of the same structure in another database. For example, in a database containing documents about different fabrics, the concept *thread* refers to a thin fibrous strand twisted together. The same concept within a computer science document collection is defined as a mini process. It should be evident that the concept *thread* has relationships to different concepts in the two databases.

For this reason, an ECH must be developed specifically for different domains. Some semi-automatic and automatic techniques have been proposed for generating a concept hierarchy or structures similar to a concept hierarchy [CC94,HF94, SI97, SM83]. To date, however, the most successful implementations are those focusing on numeric or categorical data.

The ABI/Inform information retrieval system contains a robust thesaurus that has been developed using subject indexing terms. Associated with each concept in this thesaurus is a list of broader, narrower and related terms. We use this information to generate our ECH. Since the set of concepts in the thesaurus was identified by a domain expert, it can be viewed as a relevant set of concepts for this document collection.

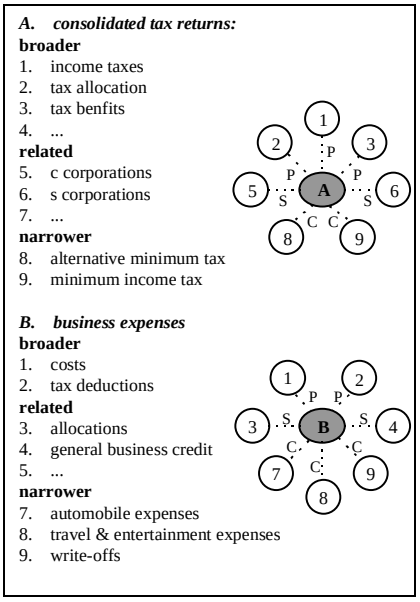

Figure 6 : Concepts & Relationships

Figure 6 shows an example of thesaurus entries for two concepts and their translation into an ECH. All of the three types of links previously defined are illustrated – parent (P), child (C), and sibling (S). We see that the data is best represented using a graph or network model. Since we are not guaranteed a single root or a single parent for each concept, a tree structure is not a suitable structure. However, because connections between concepts do have a direction, hierarchical segments exist within the network. Therefore, we define this structure to be an 'extended' concept hierarchy (ECH).

Because the ECH is being used for rule generation, efficient retrieval is of primary concern. Consequently, we store the concepts in a dynamic hash table. Although deletions to the hash table are highly irregular, updates are not. Over time, documents and concepts will continue to be added to the collection. For this reason, a static hash table may lead to a deterioration in retrieval performance. We, therefore, use linear hashing since it

1. Get document ids of documents containing structured data value, S_1, using SQL statement. (Set A)
2. Get document ids of documents containing unstructured concept, C_1, from ECH. (Set B)
3. Find intersection of sets A & B. (Set C)
4. Get document ids of concept C_r , where C_r is related to C_1 via edge P,C, or S if the support between concepts C_r and C_1 is above the minimum support. (Set D)
5. Find intersection of sets C & D. (Set E)
6. Confidence of rule = # of elements in E / # of elements in C.
7. If confidence of rule > minimum specified confidence, return rule.
8. Repeat for each unvisited concept C_r.

Figure 7 : Basic Rules Generation Algorithm

dynamically modifies the hash function as the hash table grows [LIT80].

For each concept entry in the hash table, we maintain pointers to a document id table and a relationship table. A list of documents containing the concept is stored in the document id table. The relationship table maintains a list of related concepts, the relationship type for each concept (P,C,S), and the *support* or weight of each relationship. As described in section 3.2, the support of each relationship is determined by the subset of documents two related concepts have in common. As an example, if concept C_1 appears in 500 documents and concept C_2 appears in 600 documents, 100 of which concept C_1 also appears in, then the support of their relationship is 100 / 1000 or 0.1. If the minimum support is determined to be 0.2, no rules using the relationship between concept C_1 and concept C_2 will be generated.

Since each concept is unique, using it as the key to an entry was our first choice. We define a string to be all the terms comprising a concept. However, some of the strings are fairly lengthy, and storing them would be a waste of space. Therefore, we manipulate the characters in each string to create a key for each entry in the table. This approach is a known variation of the "scrambling" process usually performed in hashing schemes to generate a key from a pseudo-key. It has the additional property that the encoding scheme is reversible, thereby allowing us to reconstruct the original key. Additional space and time improvements can be obtained by implementing a bit indexing scheme instead of maintaining actual pointers to the related document ids. This idea is similar to the one proposed in [OG95].

5.3 Mining with an Extended Concept Hierarchy

Given the document database, we manually create an ECH from the subject index terms assigned to the data. Each index term represents one or more unstructured blocks of text. The final ECH represents relationships between concepts and serves as an index to the document collection. The ECH is a compact structure that is easily expandable. As previously mentioned, once this structure is imposed, any data mining tools that require structured information to extract patterns will be able to associate data from different tables in the database to the ECH.

As previously mentioned, a user provides an unstructured concept, *finance*, a structured component, *Harvard Business Review*, and a minimum specified confidence. Using an ECH, the data mining tool needs to find the intersection between the documents published in *Harvard Business Review* and the documents containing the concept *finance*. Then the data mining tool checks the different relationships *finance* has to other concepts in the ECH. If another concept, i.e. *taxation*, has a strong relationship to *finance*, then a rule using it can be generated. If the rule's confidence is greater than the minimum specified confidence, it is returned as an association rule. Figure 7 outlines the basic rules generation algorithm. We generate four types of rules for each concept: sibling, parent, child and general rules. The semantics of the final rule vary depending upon the concept relationship type used, i.e. parent (P), child (C), or sibling

General Rule

Structured Author Value : Joe Smith

Original Concept : taxation

1. 10% of Joe Smith's articles involve taxation.

Parent Rules

Structured Journal Value : Inc. Magazine

Original Concept : tax returns

Parent Concepts : income taxes, tax planning, tax allocation

2. Articles involving tax returns typically generalize to tax planning.
3. 30 % of tax planning articles related to tax returns appear in Inc.
4. Tax return articles appearing in Inc. generally fall into the category of income taxes : 44%

Child Rules

Structured Location Value : California

Original Concept : business expenses

Children Concepts : sales tax, write-offs, bad debt

5. The majority of documents involving write-offs generalize to business expenses.
6. 10 % of bad debt articles related to business expenses involve California.
7. Business expense articles about California focus on sales taxes.

Sibling Rules

Structured Author Value : Joe Smith

Original Concept : business forecast

Sibling Concepts : business cycles, business indicators, business conditions

8. Together business forecast, business cycles, business indicators and business conditions identify 50 % of the articles written by Joe Smith.
9. More documents about business cycles are written by Joe Smith than articles about business forecasts.

Figure 8 : Rules Grouped by Concept Relationship Type

(S). Figure 8 shows examples of different rules grouped by the concept relationship type.

The rules presented are English sentences. We are able to accomplish this because the number of different structured attributes is limited. Specifically, we use only author, location and publication information. If our database contained a large number of structured values, finding generic language that can accurately express meaningful final rules becomes a difficult problem. The remainder of this section explains the variations of the basic algorithm necessary to generate each rule. The discussion is based on the examples of rules shown in Figure 8.

General Rule: The general rule is a simple association between the specified concept and the specified structured component. Only steps 1 through 3 of the rules generation algorithm are needed to generate the general rule. The confidence of the rule is the ratio between the number of element in Set C and Set A:

$$\frac{\text{\# of elements in Set C}}{\text{\# of elements in Set A}}$$

Parent Rules: We generate three different parent rules using parent concepts of the original concept specified by the user. The rules attempt to associate a structured attribute value, the original concept, and parents of the original concept. Rules 3 and 4 are quantitative in nature, while Rule 2 is qualitative. Rule 2 is determined by finding the parent concept with the largest weighted relationship, support, to the original concept. Rule 3 follows the basic rules generation algorithm. For Rule 4, during step 7, rules are not returned. After step 8, the parent concept producing the highest confidence is returned to the user.

Children Rules: Similarly, we generate three types of children rules by attempting to find relationships between the original concept, the children concepts of the original concept, and the structured attribute value. Rules 5 and 7 are qualitative in nature, while Rule 6 is an example of a quantitative rule. Similarly to Rule 2, Rule 5 is determined by finding the child concept with the largest weighted relationship, support, to the original concept. Rule 6 follows the basic rules generation algorithm, while Rule 7 only returns the child with the highest confidence.

Sibling Rules: Finally, we generate two different sibling rules. Since siblings are typically viewed as synonyms, it is interesting to determine how well these concepts describe documents associated with specified structured values. Once again, we generate both qualitative and quantitative rules. Rule 8 is a qualitative rule that attempts to categorize the "merged" concept composed of the original concept and its siblings with respect to the structured attribute value. The major deviation from the basic rules generation algorithm is step 2. In that step, the document ids of the original concept and all the sibling concepts must be obtained from the ECH. Rule 9 is a discriminant association rule which compares the original concept and its strongest sibling with respect to the structured attribute value. To generate this rule, step 5 and 6 of the basic rule generation algorithm need to be modified:

5. Find the intersection between sets A & D. (Set E)
6. If set E is larger than set C, return the rule.

The parent rules and the children rules involve essentially the same calculation. The only distinctions are the type of relationship extracted from the concept hierarchy and the grammatical construction of the final rule. Also notice that the general rule can be constructed for any of the parent, children or sibling concepts. The most important observation about the rules in Figure 9 is that Rules 2 through 9 could not have been generated without the relationship information stored in the ECH. Not only does the ECH tell us that two concepts are related, but it also specifies the weight and type of relationship that exists between the two concepts.

6 Performance Results

In this section, we show performance results that indicate the efficiency of the rule generation approach described in the previous section. We begin by analyzing the time complexity of the data mining algorithm. Specifically, we show that the time it takes to generate different groups of rules is linear with respect to the product of the number of relationships a concept has and the number of documents in the database. Our experiments confirm this upper bound.

6.1 Analysis of Time Complexity

The cost of generating these association rules can be broken down into three major components:

1. The pre-processing cost for constructing the concept hierarchy
2. The cost of getting the structured data value and the associated document ids from the database.
3. The cost of getting the concepts and their corresponding document ids from the ECH.

Relationship Type	# of Relationships		
	1	10	100
PARENT	0.0095	0.0952	0.8567
CHILD	0.0010	0.0976	0.9413
SIBLING	0.0270	0.3204	3.9740

Table 2 : Running Times of Association Rules in seconds (D = 1000)

4. The cost of finding the intersection and union of different sets of document ids.

For this analysis, we ignore the first cost. Construction of the concept hierarchy is an off-line operation that was a one time cost for this document database. We factor out the second cost since any data mining algorithm proposed would endure this cost. We realize that this cost can in fact be the dominant cost if a large number of document ids need to be found; however, we are interested in determining the scalability of using an ECH within the data mining procedure.

The next cost involves accesses to the ECH. Since we are using a linear hash table, on average, 1.5 accesses are necessary to extract a concept from the ECH [LIT80]. The document ids related to a concept are stored in a different file. For a particular concept, all of the document ids will fit into one or two pages. This implies, a constant number of accesses is needed to get document id information from the ECH. Therefore, step 3 has a constant cost.

The cost associated with step 4 is calculated as follows. In order to generate any of the rules described in the previous section, we need to scan and compare at most R lists of document ids, where R is the number of relationships a concept has. Each list of document ids contain O(D) elements, where D is the number of documents in the collection. Therefore, the total cost of step 4 is O(R*D).

6.2 Experiments

This section discusses the running times associated with developing the different rules in Section 3. The timing results for two different tests are shown in Table 2. In each test, a different initial concept and structured value are input. For the results in this table, the number of documents associated with a concept or a related concept is set to 1000. In this manner, we can more clearly determine the effect of increasing the number of relationships.

The rules are grouped into three categories: parent, child and sibling. Note that the time associated with the retrieval of the structured data in the database is not shown since we are interested in evaluating the cost related to the generation of rules using the ECH. From Table 2, we confirm that as the number of relationships, R, increases and the number of documents, D, remains constant, the time needed to generate a set of rules increases linearly with R.

Relationship Type	# of Documents		
	100	1000	10,000
PARENT	0.131	0.857	8.133
CHILD	0.135	0.941	12.699
SIBLING	2.061	3.974	60.558

Table 3 : Running Times of Association Rules in seconds (R = 100)

Table 3 keeps the total number of relationships, R, constant at 100 while increasing the number of documents, D. These running times confirm that as D increases and R remains constant, the time needed to generate a set of rules increases linearly with D. Therefore, our results do confirm the running time established in section 6.1.

7 Further Research

To date, very little work has been done in the area of mining semi-structured data. Much work still remains. This section describes some potential applications of this work and areas of further research. Specifically, more work needs to be done to investigate ways to automate the creation of the ECH for non-numeric, non-categorical data. This work uses indexing terms as concepts in the ECH. An improvement would be to also use terms extracted directly from the unstructured text itself. By doing this, a larger number of relevant concepts can be assigned to each document. This in turn enables us to identify more complex and robust associations that appeal to a larger audience.

Another area of improvement involves incorporating multiple structured values into a single rule. The rules described in this paper only involve one structured value. Within the context of the ABI/Inform document database, it would be interesting to find associations that involved document concepts from the ECH, author, and publication.

The remainder of this section describes possible applications of this work on the WWW or in conjunction with digital libraries.

7.1 Extensions for WWW

Researchers are developing more sophisticated tools that search for relevant data on the web. However, suppose we are interested in extracting patterns and associations that exist within a WWW site, as well as across WWW sites. Without the assistance of data mining tools, extracting patterns from such a large data set would take exponential time. Since databases are defined as repositories for storing information or data, we can view the Internet as an ill-structured database. Each web page or set of web pages can be viewed as a document that contains structured and unstructured components. If we focus on the textual information, the WWW can be viewed as a huge document database. Consequently, the data association approach described in this paper could be extended to extract patterns from HTML pages on the Internet.

However, in order to implement this approach, a few questions need to be addressed:

1. Where can we find the concepts for the ECH?
2. Once we find the concepts, how do we determine which concepts relate to each other?
3. How can we associate web sites, documents, with concepts in the ECH?
4. Where will store this ECH? What data structure should be used?

Issues 1, 2, and 3 can be tackled simultaneously by making use of information that search engines currently provide their users. The major search engines developed for the WWW have large concept lists. These lists are synonymous with index terms in a document database. Concepts from numerous search engine hierarchies can be combined to form the backbone of the ECH. Relationships among concepts are also identified by the major search engines. Further, since web sites are assigned to each concept, we should be able to use this information to associate web sites with concepts in our ECH.

As an example, suppose Joe Smith is opening a flower shop in Chicago, and he decides that he wants to sell flowers not found in other flower shops in the area. The following rules could influence his choice of flowers:

5% of flower shops in Chicago sell lilies.
80% of flower shops in Chicago sell roses.

If WWW pages owned by flower shops contain a listing of flower types, this type of information can be extracted using the association rules described in Section 3. Other information that can be extracted in this manner includes pricing data, corporate profile data, and sport preference associations. [DEW96] investigate using an intelligent agent to automatically compare prices of a product sold by numerous vendors on the WWW. Once this agent has successfully found the cheapest product, our data mining algorithm can be used to identify patterns of different vendors. As an example, we might find that Vendor X typically has sales on computers once a month. In this manner, these rules can be used as resource discovery tools to compare information on different sites or to initiate actions in decision support systems.

Because the WWW is very dynamic, updates to the ECH may be frequent. Therefore, the data structure chosen for storage of the ECH must allow for efficient updates and retrieval. The dynamic linear hash table employed for our ECH is a nice option. Some other options include creating relational tables or using an object oriented concept model.

7.2 Increasing Functionality of Digital Libraries

Currently, digital libraries provide much of the same functionality as information retrieval systems within libraries. Techniques for intelligently gathering documents from the web are being investigated. Some approaches include intelligent agents or robots and natural language processing of documents. The document collection examples presented in this paper can be expanded to generate rules that associate document data from different collections across the web. One or more ECHs can be used to associate the information. Issues that arise include inconsistencies between ECHs representing different document collects and different representation of structured data values within the document collection.

7.3 Concluding Remarks

This paper presents a new and efficient method for relating information from structured and unstructured portions of a document database. It accomplishes this task by using a pre-generated extended concept hierarchy to identify and relate concepts that appear in the unstructured components of the documents. The association algorithms then generate qualitative and quantitative rules about subsets of documents containing user specified structured data values and unstructured concepts.

We propose an efficient implementation of the extended concept hierarchy based on linear hashing that keeps track of concepts, pointers to documents related to them, and relationships among concepts. Using a subset of the ABI/Inform document collection, we present experimental results that confirm the effectiveness of our approach. Finally, we conclude by describing extensions of our work for the WWW domain and digital libraries.

References

[AIS93] R. Agrawal, T. Imielinski, & A. Swami, "Mining Association Rules Between Sets of Items in Large Databases," In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, May 1993.

[AS94] R. Agrawal & R. Srikant, "Fast Algorithms for Mining Association Rules," In *Proceedings of the International Conference of Very Large Databases*, September 1994.

[CC94] W. Chu & K. Chiang, "Abstraction of High Level Concepts from Numerical Values in Databases," In *AAAI Workshop on Knowledge Discovery in Databases*, July 1994.

[DEW96] R. Doorenbos, O. Etzioni, D. Weld, "A Scalable Comparison-Shopping Agent for the World-Wide Web Domain," Technical Report 96-01-03, University of Washington, Department of Computer Science and Engineering, January 1996.

[FD95] R. Feldman & I. Dagan, "Knowledge Discovery in Textual Databases." In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 1995.

[FL96] S. Fortin & L. Liu, "An Object-Oriented Approach to Multi-Level Association Rule Mining," In *Proceedings of the International Conference on Information & Knowledge Management*, 1996.

[FSSU95] U. M. Fayyad, G Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1995.

[HF94] J. Han & F. Yongjian, "Dynamic Generation and Refinement of Concept Hierarchies for Knowledge Discovery in Databases." In *AAAI Workshop on Knowledge Discovery in Databases*, July 1994.

[HF95] J. Han and Y. Fu, "Discovery of Multiple-Level Association Rules from Large Databases," In *Proceedings of the International Conference of Very Large Databases*, September 1995.

[KMR94] M. Klemettined, H. Mannila, P. Ronkainen, H. Toivonen, & I. Verkamo, "Finding Interesting Rules from Large Sets of Discovered Association Rules," In *Proceedings of the International Conference on Information and Knowledge Management*, November 1994.

[KS94] V. Kashyap & A. Sheth, "Sematics-based Information Brokering," In *Proceedings of the Third International Conference on Information and Knowledge Management*, Nov. 1994.

[LHKK96] K. Lagus, T. Honkela, S. Kaski, & T. Kohonen. "Self-Organizing Maps of Document Collections: A New Approach to Interactive Exploration," In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 1996.

[LIT80] W. Litwin. "Linear Hashing: A New Tool for File & Table Addressing," In *Proceedings of the International Conference of Very Large Databases*, 1980.

[OG95] P. O. O'Neil & G. Graefe, "Multi-Table Joins Through Bitmapped Join Indices," *SIGMOD Record*, 24(3), September 1995.

[PBKK96] Piatetsky-Shapiro, Gregory, Ron Brachman, Tom Khabaza, Willi Kloesgen, and Evangelos Simoudis, "An Overview of Issues in Developing Data Mining and Knowledge Discovery Applications." In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 1996.

[PCY95] J. S. Park, M. S. Chen, P. S. Yu, "An Effective Hash-Based Algorithm for Mining Association Rules," In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, May 1993.

[SA95] R. Srikant & R. Agrawal, "Mining Generalized Association Rules," In *Proceedings of the International Conference of Very Large Databases*, September 1994.

[SI97] L. Singh, Automatic Preprocessing & Transformation of Semi-Structured Data for Data Mining Applications, May 1997.

[TPL95] M. Trensh, N. Palmer and A. Luniewski, "Type Classification of Semi-structured Documents." In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1995.