Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Naveen Tirupattur

Entitled
TEXT MINER FOR HYPERGRAPHS USING OUTPUT SPACE SAMPLING

For the degree of Master of Science

Is approved by the final examining committee:

Snehasis Mukhopadhyay
Chair

Shiaofen Fang

Yuni Xia

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Snehasis Mukhopadhyay

Approved by: Shiaofen Fang 04/07/2011
Head of the Graduate Program Date

Graduate School Form 20
(Revised 9/10)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:

TEXT MINER FOR HYPERGRAPHS USING OUTPUT SPACE SAMPLING

For the degree of Master of Science

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22,* September 6, 1991, *Policy on Integrity in Research.**

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Naveen Tirupattur

Printed Name and Signature of Candidate

04/08/2011

Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

TEXT MINER FOR HYPERGRAPHS USING OUTPUT SPACE SAMPLING

A Thesis

Submitted to the Faculty

of

Purdue University

by

Naveen Tirupattur

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

May 2011

Purdue University

Indianapolis, Indiana

To,

Avva

ACKNOWLEDGMENTS

I would like to express my deep and sincere gratitude to my advisor, Dr. Snehasis Mukhopadhyay for his guidance and encouragement throughout my Thesis and Graduate studies.

I also want to thank Dr. Shiaofen Fang and Dr. Yuni Xia for agreeing to be a part of my Thesis Committee. I thank Dr. Mohammed Al Hasan for providing me his guidance during various stages of my Thesis work. I thank Dr. Joseph Bidwell for his inputs and feedback on protein data.

Thank you to all my friends and well-wishers for their good wishes and support. And most importantly, I would like to thank my family for their unconditional love and support.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Tirupattur, Naveen. M.S., Purdue University, May, 2011. Text Miner for Hypergraphs using Output Space Sampling. Major Professor: Snehasis Mukhopadhyay.

Text Mining is process of extracting high-quality knowledge from analysis of textual data. Rapidly growing interest and focus on research in many fields is resulting in an overwhelming amount of research literature. This literature is a vast source of knowledge. But due to huge volume of literature, it is practically impossible for researchers to manually extract the knowledge. Hence, there is a need for automated approach to extract knowledge from unstructured data. Text mining is right approach for automated extraction of knowledge from textual data. The objective of this thesis is to mine documents pertaining to research literature, to find novel associations among entities appearing in that literature using Incremental Mining. Traditional text mining approaches provide binary associations. But it is important to understand context in which these associations occur. For example entity A has association with entity B in context of entity C. These contexts can be visualized as multi-way associations among the entities which are represented by a Hypergraph. This thesis work talks about extracting such multi-way associations among the entities using Frequent Itemset Mining and application of a new concept called Output space sampling to extract such multi-way associations in space and time efficient manner. We incorporated concept of personalization in Output space sampling so that user can specify his/her interests as the frequent hyper-associations are extracted from the text.

CHAPTER 1. INTRODUCTION

Advancements in computer science have made the access to information very easy for the researchers. Literature is an important source of information for any researcher during course of study on a research problem. There is abundance of literature to access for any researcher due to rapid growth of online tools. This abundance of information/literature is overwhelming for the researchers. Due to sheer volume of literature, it is impossible to extract all the knowledge from it. There is also possibility of misinterpretation. Hence there is need for automated knowledge extraction from large amount of data. Due to availability of literature in machine readable format has led to development of automated approaches like text mining possible.

Text mining [1] which is based on Natural Language Processing [2] and Artificial Intelligence [3] , is challenging because the data is unstructured in many cases i.e. textual data in the literature does not follow a fixed hierarchy to allow easy extraction of meaningful information. It becomes even more challenging when multiple objects and multiple associations need to be extracted. But, it is a promising approach with a high potential to extract knowledge contained in research literature. Because the most natural form of storing and communicating information is in text format.

Natural language processing has wide range of applications including translating information from machine readable format to human readable format and vice versa into data structures or parse trees etc. NLP is closely associated with artificial intelligence. Artificial Intelligence also known as AI can be defined as “study and design of intelligent systems”. Machine learning and unsupervised learning are two broad categories in AI. It has wide variety of applications in field of computer science including medical diagnosis, stock trading, robot control, law, scientific discovery and toys.

Textual data is used to extract novel associations among the entities appearing in the literature using text mining approaches. These associations are verified later by experiment. The association strengths are calculated based on co-occurrence of entities in the literature. Typical steps involved in extraction of associations by text mining are: document extraction, document representation, weight computation for entities and finally score computation for associations among the entities. This thesis work uses well known TF-IDF algorithm [4] for assigning scores to the entity associations. Traditional text mining approaches extract binary associations among the entities. In some scenarios, it is imperative that context in which these associations occur also be extracted from text for better understanding of the associations. These contexts can also be entities appearing in the literature, thus there is a need for multi-way association extraction from textual data.

Traditional text mining approaches start with set of entities of interest and extract all the documents which contain these entities. Following this, text mining is done on the documents in the dataset to extract co-occurrence based associations between each pair of these entities. These associations are assigned a score and all the associations with scores above a predefined acceptance score are filtered for further verification by experiment. These approaches have been proven to yield novel associations but they are not efficient when the data size is huge. In this thesis we propose an incremental mining approach, which does text mining in incremental steps building upon the work done during the preceding iterations. There has been considerable research work on finding binary associations from textual data but little work on finding multi-way associations. Multi-way associations assume significance in situations where there is a need to understand the context of association along with association. For example instead of extracting a simple binary association of “protein A interacts with protein B” it makes much more sense to understand “protein A interacts with protein B in domain C under influence of drug D”. These multi-way associations can be represented by a hypergraph [5] with edges representing multi-way associations among the vertices which are entities co-occurring in the literature. A hyper edge can connect more than two vertices at a time thus suitable for representing multi-way associations effectively. Normal graph can be considered as a special case of hypergraph connecting two vertices by an edge.

These multi-way associations can be extracted from textual data using frequent itemset mining (FIM). FIM also known as association rule learning [6] is a popular and well researched method for discovering interesting relations between variables in large databases. FIM is employed in market basket analysis and in many application areas including web usage mining, intrusion detection and bioinformatics. A hypergraph is a generalization of a graph with edges connecting more than 2 vertices. Formally, a hypergraph H is a pair $H = (X, E)$ where X is a set of *vertices*, and E is a set of non-empty subsets of X called *hyper edges*. $X = \{v_1,v_2,v_3,v_4,v_5,v_6,v_7\}$, $E = \{e_1,e_2,e_3,e_4\} = \{\{v_1,v_2,v_3\},\{v_2,v_3\},\{v_3,v_5,v_6\},\{v_4\}\}$. So, each edge of a hypergraph can represent a relationship between more than two objects. Hyper edges represent the multi-way associations (hyper-associations) occurring among the entities in the literature. This thesis work focuses on extracting such hyper-associations based on co-occurrence of entities in textual data.

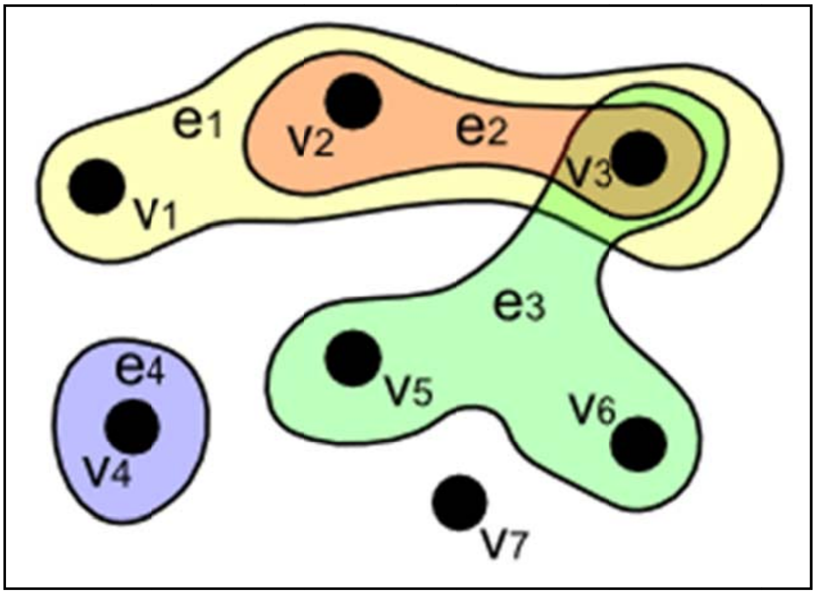

Figure 1 Sample hypergraph

The objective of this thesis is to develop an Incremental mining approach to extract associations among the entities appearing in research literature. Thus extracted associations were assigned a score to show the strength of the association using TF-IDF algorithm based on principle of co-occurrence. This thesis also discusses extracting frequently occurring multi-way associations among the entities in the literature using FIM techniques like Apriori [24] and ECLAT (Equivalence CLAss Transformation) [28] algorithms. We also propose a novel approach called Output space sampling which is a random walk algorithm of type Metropolis-Hastings [7] based on Markov Chain Monte Carlo class of algorithms. The textual data used for mining is downloaded from PubMed [8] database which is an online repository for medical literature.

In this thesis we incorporated concept of personalization in Output space sampling to allow users to choose his/her preferences during the frequent item set mining process. In first variation of personalization user selects a set of hyper-associations he/she is interested in. The Output sampling is done only on these hyper-associations instead of all the hyper-associations extracted from documents. Hence the random walk is done only on these hyper-associations and frequent hyper-associations are presented to the user at the end of random walk.

In second variation, user continuously provides feedback to the system if he/she is interested in the hyper-association chosen by system during random walk at each level. Based on the user feedback system appropriately selects the next hyper-association from set of available hyper-associations. Thus user continuously determines what hyper-associations he/she is interested in, during the random walk at each and every level. Thus at the end of random walk user has only a distribution of hyper-associations among the hyper-associations he/she is interested.

This thesis is divided into 2 parts: incremental mining and frequent itemset mining which is further divided into 3 parts: Apriori, ECLAT and Output space sampling. Output space sampling section explains all the personalization variants we implemented in this thesis work for performing a random walk on entities extracted from the text to extract frequent multi-way associations (hyper-associations). Data for testing all these approaches was downloaded from PubMed.

CHAPTER 2. BACKGROUND

This thesis draws motivation from paper by Vaka and Mukhopadhyay [9] which describes mechanism to find novel associations among biological entities related to the ancient Indian medical practice called Ayurveda [10] and the modern biomedical literature. This thesis is an extension to work done in [9] in terms of extracting novel associations in an efficient manner. There has been significant research going on in application of text mining in research literature. One such approach by was proposed by Collier et al. [11] examines information retrieval methods for classification of entities which appear in abstracts from online medical database MEDLINE [12]. This approach uses decision tree structures for classification and entity identification. Kostoff et al. [13] describe a novel approach for identifying the pathways through which research can impact other research, technology development, and applications, and to identify the technical and infrastructure characteristics of the user population. A novel literature-based approach was developed to identify the user community and its characteristics.

There have been similar incremental mining approaches proposed in association rule mining to find frequent patterns from sequence databases. Masseglia et al. [32] present a new algorithm for mining frequent sequences that uses information collected during an earlier mining process to cut down the cost of finding new sequential patterns in the updated database. They found out that in many cases it is faster to apply their algorithm than to mine sequential patterns using a standard algorithm, by breaking down the database into an original database plus an increment. Sayed et al. proposed an incremental miner for mining frequent patterns using FS-tree [33] which has the ability to adapt to changes in users' behavior over time, in the form of new input sequences, and to respond incrementally without the need to perform full re-computation. Their system allows the user to change the input parameters (e.g., minimum support and desired pattern size) interactively without requiring full re-computation in most cases. Smalheiser [14] describes a method to connect meaningful information across various domains of research literature. The study was conducted using series of MEDLINE searches. This method defined two domains of research, assumed to contain meaningful information and to find common entities that bridge these domains. This method required lot of manual intervention by domain experts in the form of feedback to find the pathways that bridge the domains.

Transminer by Narayanasamy et al. [15] finds transitive associations among various biological objects using text-mining from PubMed research articles. This system is based on the principles of co-occurrence and uses transitive closure property for extracting novel associations from existing associations. The extracted transitive associations are given a score using TF-IDF method.

Donaldson et al. [16] proposed a system based on support vector machines to locate protein-protein interaction information in the literature. They present an information extraction system that was designed to locate interaction data in the literature and present these data in machine readable format to researchers. This system is currently limited to human, mouse and yeast protein-interaction information. EDGAR [17] is another similar natural language processing system that extracts relationships between cancer-related drugs and genes from biomedical literature.

Srinivasan [18] demonstrated an approach to generate hypotheses from MEDLINE. This paper proposes open and closed text mining algorithms that are built within the discovery framework established by Swanson and Smalheiser

[19]. This approach successfully generated ranked term lists where the key terms representing novel relationships between topics are ranked high. Swanson and Smalheiser found an association between magnesium and migraine headaches that was not explicitly reported in any one article, but based on associations extracted from different article titles, and later validated experimentally. In this approach a set of articles related to user’s topic of interest are downloaded and software generates another set of articles based on downloaded titles complementary to the first set and from a different area of research. The two sets are complementary i.e. when together they can reveal new useful information that cannot be inferred from either set alone. The software further helps the user identify the new information and derive from it a novel hypothesis which could be later verified by hypothesis. But, this approach is limited to titles of the articles/documents.

Nenadic and Ananiadou’s [20] article discusses the extraction of semantically related entities (represented by domain terms) from biomedical literature. Their method combines various text-based aspects, such as lexical, syntactic, and contextual similarities between terms. Yeganova et al. [21] describe a similar system to query gene/protein name which identifies related genes/proteins from a large list. Their system is based on a dynamic programming algorithm for sequence alignment in which the mutation matrix is allowed to vary under the control of a fully trainable hidden Markov model. This thesis work is based on the paper by Mukhopadhyay et al. [22] which discusses generation of hypergraphs representing multi-way association among various biological objects. They presented exhaustive and Apriori methods. This thesis work extends their work by using ECLAT and novel concept of Output space sampling along with Apriori approach to extract multi-way associations.

r-Finder system proposed by Palakal et al. [23] finds biological relationships from textual data. Their paper presents an approach to extract relationships between multiple biological objects that are present in a text document. Their approach involves object identification, reference resolution, ontology and synonym discovery, and extracting object-object relationships. Hidden Markov Models (HMMs), dictionaries, and N-Gram models are used to set the framework to tackle the complex task of extracting object-object relationships. But it could only find binary relationships not multi-way associations among the entities in the text.

Many algorithms have been proposed for association rule mining like Apriori [24], FP growth tree [25], OPUS search [26], OneR [27] etc. Most of the approaches are horizontal and require multiple database scans to find frequent patterns. Vertical approaches like ECLAT [28], GUHA [29], MaxEclat and Clique [28] were also proposed. Most of these algorithms are exhaustive i.e. they generate all possible k-item candidate patterns at each level k to find k+1-item frequent patterns.

Apriori [24] is best-known algorithm to mine frequent itemsets. It’s a horizontal approach which uses breadth-first strategy to count the support of itemsets and uses a candidate generation function which exploits the downward closure property of support. The support of an itemset X (support(X)) is defined as the proportion of transactions in the data set which contain the itemset. This algorithm suffers from inefficiencies like large numbers of candidate generation and bottom-up subset exploration which consume lot of time and space.

ECLAT is a vertical mining approach proposed by Zaki [28]. It also uses a breadth-first search algorithm with set intersection to find frequent itemsets. This algorithm utilizes the structural properties of frequent itemsets to facilitate fast discovery. The items are organized into a subset lattice search space, which is decomposed into small independent chunks or sub lattices, which can be solved in memory. This is a very efficient algorithm when compared to Apriori as it avoids computation expensive step of candidate set generation. FP-growth [25] uses an extended prefix-tree (FP-tree) structure to store the data in a compressed form. FP-growth adopts a divide-and-conquer approach to decompose the mining tasks. It uses a pattern fragment growth method to avoid the costly process of candidate generation and testing used by Apriori.

This thesis work implements vertical mining approach proposed by Zaki [28] to find frequent itemsets. Zaki et al. [30] proposed a fast vertical mining approach with a novel concept called diffsets. It is an extension to set intersection approach where only difference in transaction ids of candidate patterns and its generated frequent patterns are stored. This they believe will drastically reduce memory required to store intermediate results and improve performance significantly.

Hasan et al. [31] proposed a novel approach in graph pattern mining to find frequent sub graphs. His approach is a generic sampling framework that is based on Metropolis-Hastings algorithm to sample Output space of frequent sub graphs. This thesis work is an application of [31] and set intersection of [28] in text mining to find frequently appearing multi-way associations from textual data.

CHAPTER 3. METHODOLOGY

This thesis is broadly divided into two parts: incremental mining and frequent itemset mining (FIM). FIM is categorized into 3 sub sections: Apriori, ECLAT and Output space sampling. This chapter contains detailed descriptions of all work done as part of this thesis.

3.1. Incremental Mining

In this thesis work we propose a novel approach called incremental mining to efficiently extract novel associations among the entities appearing in research literature. In contrast to several traditional text mining approaches, our approach is computationally efficient and accurate. This process has 4 major tasks:

1. Document Extraction
2. Document Representation
3. Weight Matrix Computation
4. Association Matrix Computation

Abstracts containing the entities of interest are downloaded from PubMed. The data from these abstracts is represented in a suitable format for further processing. Association strengths between entities are calculated using TF-IDF algorithm. Based on a pre-defined threshold value all the scores below this threshold are filtered out and remaining associations are verified by experiment by domain experts.

3.1.1. Data Extraction

The process of document extraction begins with querying PubMed with set of entities. The query returns the document ids of all the documents in which these entities occur alone or together. Then PubMed is queried again with the document ids obtained in previous step. The query returns a XML response which is parsed to extract the text in each document. The text from each document is written to a separate file with file name as document id which will be used in next step for data extraction. The incremental approach in this step is we store the document ids returned in previous iteration in a file so when user runs it next time with new entities added to original list; only documents whose ids were not in original document id list are downloaded. This change enhances the performance.

3.1.2. Document Representation

The data extracted from all the documents must be represented as some data structure which captures the document information and the entity information appearing in that document. The ideal data structure which matches our requirement is a Map. Map is a data structure which stores data as a <key, value> pairs. We construct a document map which stores the document id as key and another map (which has entity name as key and its frequency in this document as value) as value in the first iteration of incremental miner. A significant performance improvement was found by changing the document representation to store only the entities whose frequency is non-zero in the document map. This modified document representation was used in this thesis work. This document map is then converted a matrix which has documents as rows and entities as columns with matrix elements storing the frequency of each entity in that document. This matrix is called as Term Frequency matrix (TF). The document map constructed is then written to a file and during the next iterations of incremental miner, this file is read and document map is recreated. So, we have the document representation of all documents and entities appearing in those documents. Now only the new documents and new entities are added to this map. This change saves time needed for creating a document map during iteration. The document format is shown in table 1.

Table 1 Document representation format in Incremental Miner

Document Id	Entities in document
1	<A,1>, <C, 2>, <E, 1>, <F, 4>
2	<C,4>, <D, 2>, <F, 4>
3	<A, 7>, <C, 3>, <E, 9>, <F, 1>
4	<A, 6>, <C, 4>, <D, 1>, <F, 3>
5	<A, 3>, <C, 2>, <D, 4>, <E, 1>
6	<C, 1>, <D, 4>, <E, 9>

3.1.3. Weight Matrix Computation

The TF matrix obtained in the previous step is used to calculate weight of each entity in that document. The weight calculation is done using the well-known TF-IDF method [2]. This weight is a statistical measure used to evaluate importance of an entity in that document to total collection of documents. The importance increases proportionally to the number of times an entity appears in the document but is offset by the frequency of the entity in the collection of documents. This formula is applied to achieve a refined distribution at the entity representation level. The inverse document frequency (IDF) component acts as a weighting factor by taking into account inter-document entity distribution, over the complete collection of documents. The weight of an entity in a document is calculated as:

$$W_{ik} = T_{ik} * \log(N/n_k) \tag{1}$$

Where Wik represents weight of entity k in document i, Tik represents frequency of entity k in document i, this value can be obtained by looking up in TF matrix, N is total number of documents and nk represents number of documents having the entity k. The weight matrix thus computed is used in calculating association matrix in next step. By using TF-IDF formula we get a normalized distribution of weights of each entity across the collection of documents.

3.1.4. Association Matrix Computation

From the previous step we get a weight matrix Wik which can be described as collection of N dimensional vectors for all documents M. N is total number of entities. The goal of this method is to find associations among the entities appearing in the collection of documents. We find pair wise associations for all the entities by multiplying weights of each entity and summing it up over all the documents. Once this process is done we have an association matrix which has pair-wise associations. The association matrix is computed as follows:

$$a[j][k] = \sum_{i=1}^{M} W_{ij} * W_{ik}\, , j = 1,2 \ldots \ldots N, k = 1,2 \ldots \ldots N \tag{2}$$

The values in association matrix represent the strength of association for that pair of entities. If a pair of entities occurs together at least once in any of the documents the corresponding value in association matrix will be non-zero. We can deduce that higher the value in association matrix, higher the degree of association between the pair of entities. Thus at end of an iteration we have a matrix with associations strength values between pairs of entities.

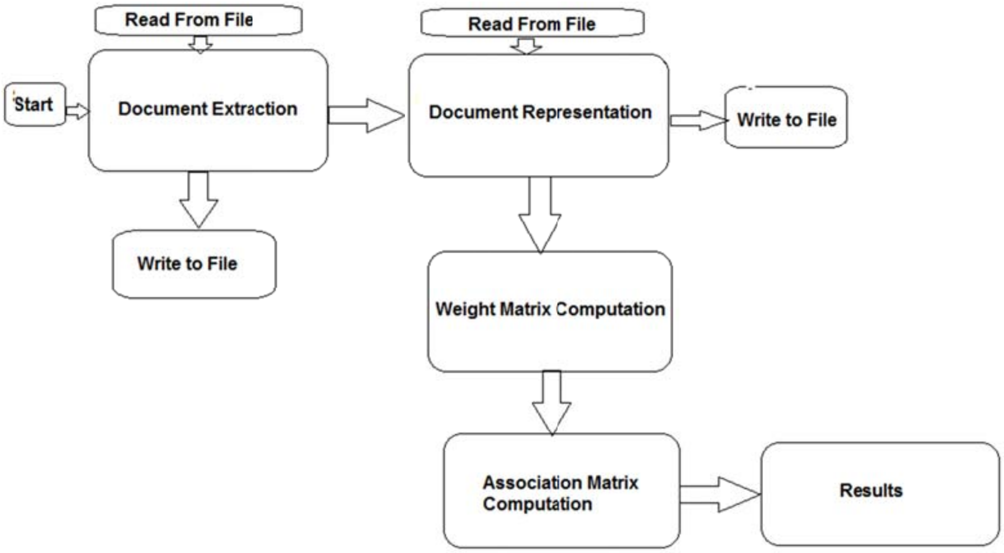

Figure 2 Incremental Mining process

3.2. Frequent Itemset Mining

3.2.1. Apriori

Apriori is a bottom-up, breadth first approach which generates frequently co-occurring entities. These co-occurring entities can be represented by a hyper-association. This process begins by generating all candidate hyper-associations of length 1 of which only frequent hyper-associations are selected for next pass which generates candidate hyper-associations of length 2 and this process continues till no more candidate hyper-associations can be generated. It achieves performance gain by reducing the size of candidate hyper-associations at each pass by filtering out the infrequent hyper-associations. Apriori is based on principle that states “All subsets of a frequent pattern are also frequent”.

The drawback of this approach is it needs to make multiple passes over the data to find the support of a hyper-association at each level. This could lead to considerable performance overhead if the hyper-association size is long because algorithm performs as many passes over the data as the length of longest frequent hyper-association. Another disadvantage is generating long candidate hyper-associations. The candidate hyper-associations that need to be generated are 2^m – 2 for a frequent hyper-associations of size m and each of which have to be examined by the algorithm to determine the frequent hyper-associations which is CPU intensive. Hence it is not efficient approach when the dataset is huge and dense. This approach is shown in figure 3. Apriori approach involves 4 major steps:

1. Document Extraction
2. Document Representation
3. Candidate k Hyper-Association Generation
4. Frequent k Hyper-Association Generation

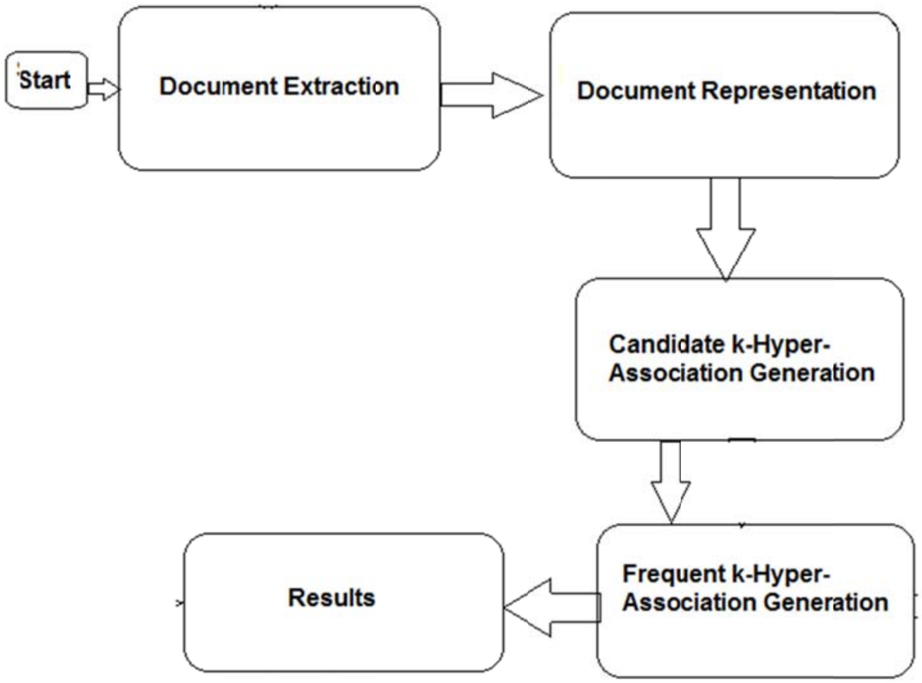
Start
Document Extraction
Document Representation
Candidate k-Hyper-Association Generation
Frequent k-Hyper-Association Generation
Results

Figure 3 Apriori algorithm

1. Document Extraction

In this step we download the abstracts containing the entities of interest from PubMed. PubMed is queried with set of entities which returns the response in XML format. The abstracts are referred to as documents from here on. The XML response contains the document-ids of abstracts which contain one or more of these entities. The PubMed is queried again with document-ids from previous step to download the abstracts. This request returns an XML response which is parsed to extract text containing in the abstract. The text from each document is written to a file having document-id as file name. These downloaded abstracts are used in next step to create a data structure which captures all the required information from textual data of abstracts.

2. Document Representation

Document representation format is critical factor in performance of any FIM approach. The data from documents are represented in a horizontal format shown in table 2. The data extracted from all the documents must be represented as some data structure which captures the document information as well the entities information appearing in that document. The ideal data structure which matches our requirement is a Map. Map is a data structure which stores data as a <key, value> pairs. Each document is stored with document id as key and entities appearing in that document as values for that key. The count of documents containing entity A is the support for entity A.

Table 2 Document representation format in Apriori

Document Id	Entities in document
1	A, C, E, F
2	C, D, F

3	A, C, E, F
4	A, C, D, F
5	A, C, D, E, F
6	C, D, E

3. Candidate k Hyper-Association Generation

In this step all candidate hyper-associations containing k entities are generated. In the first iteration i.e. k = 1, each entity is a candidate. All the entities with support less than minimum support are filtered out and remaining associations are passed to next step. These entities are stored in level 1 entity list. From next iterations, frequent k - 1 hyper-associations are used as candidates for generating candidate k hyper-associations. Each k - 1 hyper-association is joined with an entity from level 1 entity list to create candidate k hyper-association. These k hyper-associations are passed to next step to extract frequent k hyper-associations. Thus at each step of candidate generation only frequent associations from previous step are considered based on Apriori principle mentioned above.

4. Frequent k Hyper-Association Generation

Support of each candidate k hyper-association, generated in previous step is calculated by making multiple passes over the data to find the count of documents containing all the k entities occurring in this hyper-association. All the candidate k hyper-associations with support less than minimum support are filtered and remaining hyper-associations which are frequent are passed as candidates for generating candidate k+1 hyper-associations. This is repeated till there are no more frequent hyper-associations.

3.2.2. ECLAT

In this paper we used ECLAT algorithm proposed by Zaki to compare with our Output space sampling approach. He proposed a novel format to represent to data for efficient extraction of frequently co-occurring entities. It is called vertical tid-list where each entity is stored as key and documents in which it occurs as values for this key. This is contrast to traditional format where each document is stored as key and entities in it as values for that key. ECLAT also employs bottom-up breadth-first search approach to find frequently co-occurring patterns in the documents. The main steps involved in ECLAT are:

1. Document Extraction
2. Document Representation
3. Candidate k Hyper-Association Generation
4. Frequent k Hyper-Association Generation

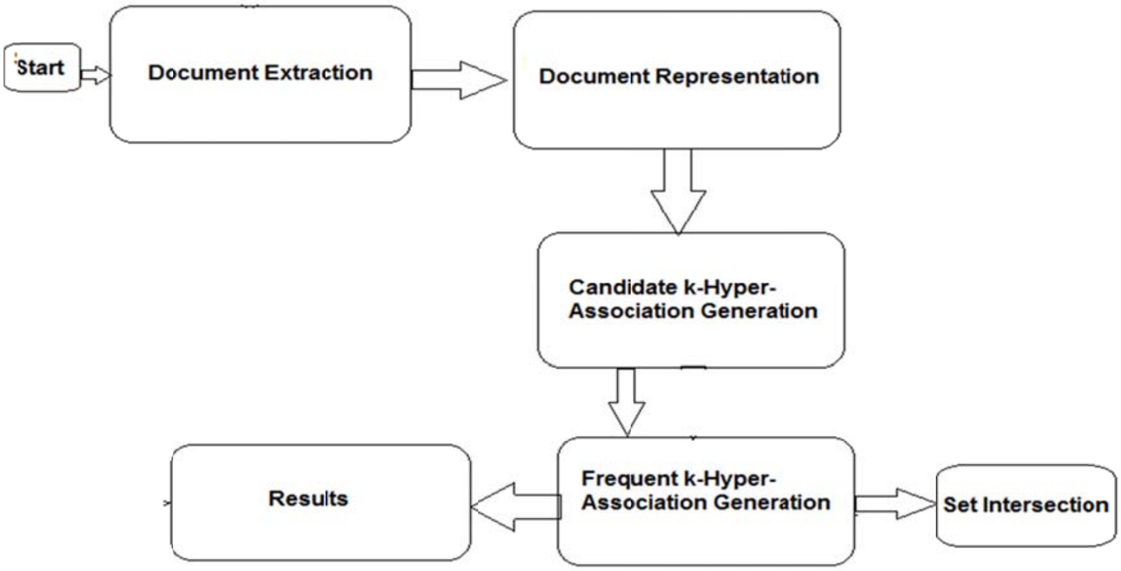
Start
Document Extraction
Document Representation
Candidate k-Hyper-Association Generation
Frequent k-Hyper-Association Generation
Results
Set Intersection

Figure 4 ECLAT algorithm

1. Document Extraction

In this step we download the abstracts containing the entities of interest from PubMed. PubMed is queried with set of entities which returns the response in XML format. The abstracts are referred to as documents from here on. The XML response contains the document-ids of abstracts which contain one or more of these entities. The PubMed is queried again with document-ids from previous step to download the abstracts. This request returns an XML response which is parsed to extract text containing in the abstract. The text from each document is written to a file having document-id as file name.

2. Document Representation

Document representation in ECLAT is entirely different from Apriori. This representation is referred to as vertical representation. This representation also uses a Map data structure but in this representation we store entities as keys in the Map as opposed to document-ids which were keys in horizontal format. Thus each entity is stored along with document-ids of all the documents containing this entity. A sample vertical representation is shown in table 3. This representation offers a simple approach to calculate support of an entity with a single pass over the data. Support of entity A is count of number of documents containing it. It can be calculated easily by counting the number of document-ids stored as values for this entity in Map data structure.

Table 3 Document representation format in ECLAT

Entity	Documents containing this Entity
A	1, 3, 4, 5

C	1, 2, 3, 4, 5, 6
D	2, 4, 5, 6
E	1, 3, 5, 6
F	1, 2, 3, 4, 5

3. Candidate k Hyper-Association Generation

Each hyper-association with k entities is used to create new hyper-association of containing k+1 entities. In the first iteration i.e. k = 1, each entity is a candidate hyper-association. All the entities with support less than minimum support are filtered out and remaining associations are passed to next step. These entities are stored in level 1 entity list. From next iterations, frequent k - 1 hyper-associations are used as candidates for generating candidate k hyper-associations. Each k - 1 frequent hyper-association is joined with an entity from level 1 entity list to create candidate k hyper-association.

4. Frequent k Hyper-Association Generation

Support of each candidate k hyper-association, generated in previous step is calculated by computing the intersection of document-ids list of two combining hyper-associations. If the size of intersection is more than minimum support, the new hyper-association is frequent. The frequent hyper-associations containing k entities are used as candidate hyper-associations to find frequent hyper-associations containing k+1 entities. This is repeated till there are no more frequent hyper-associations.

Since this is a breadth-first approach it generates all candidate and frequent hyper-associations. But this approach has advantage over Apriori in number of passes made on data to calculate support thus minimizing I/O costs especially if documents contain many hyper-associations. These advantages arise from the fact that support counting is done based on intersection size of document-ids as opposed to multiple passes over data. Vertical formats provide natural pruning of infrequent hyper-association s as candidate generation and support counting is done in one step and infrequent hyper-associations are discarded. Set intersection process is shown in figure 5.

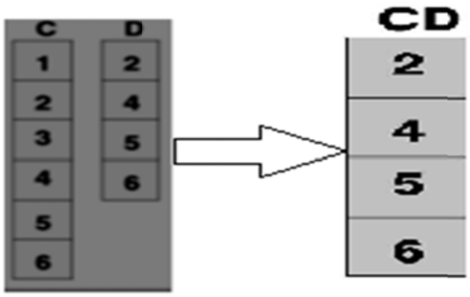

Figure 5 Set intersection

3.2.3. Output Space Sampling

In this section we describe about the Output space sampling approach we propose, in detail. The main steps of Output space sampling are:

1. Document Extraction
2. Document Representation
3. Entity Selection
4. Generate Neighbors
5. Frequent k Hyper-Association Extraction

1. Document Extraction

In this step we download the abstracts containing the entities of interest from PubMed. PubMed is queried with set of entities which returns the response in XML format. The abstracts are referred to as documents from here on. The XML response contains the document-ids of abstracts which contain one or more of these entities. The PubMed is queried again with document-ids from previous step to download the abstracts. This request returns an XML response which is parsed to extract text containing in the abstract. The text from each document is written to a file having document-id as file name. These downloaded abstracts are used in next step to create a data structure which captures all the required information from textual data of abstracts.

2. Document Representation

The textual data read from the downloaded abstracts is stored in a machine readable format for easy processing and knowledge extraction. Pattern of the entity is text contained in entity. For example if entity is “Actin”, its pattern is “Actin”. In this step all the entities are chosen one by one and for each entity chosen, all the documents are processed one by one to find the frequency of occurrence of the pattern of the chosen entity in each document. Then the document-id and frequency of the entity in each document are stored as <key, value> pairs in a map structure. This map is referred to as VAT which can store all the document-ids containing a pattern and frequency of that pattern in each document. Support of an entity is the number of documents containing this entity. An entity is represented as a data structure which can store the pattern of entity and it’s VAT. This is repeated for all entities. The support of an entity is calculated by counting the number of document-ids in VAT of that entity.

This document representation was proposed by Zaki in [5] and is called vertical data representation format. We use the same representation in our Output space sampling. VAT representation provides a direct and fast mechanism to count the support of an entity as opposed to Apriori approach where all the documents have to be processed to calculate the support. This representation improves performance significantly. Thus all the information from textual data is extracted and represented in a structure which will be used in next steps to find frequently co-occurring entities i.e. frequent hyper-associations.

3. Entity Selection

Before selecting any entity, support of each entity is checked and entities with support less than predefined minimum support are discarded. The remaining entities are added to a set called level 1 entity set. The size of a hyper-association is determined by the number of entities in that association. This is called level1 entity set because only hyper-associations of size 1 are added to this set. We start with only frequent hyper-associations of size 1. In the first iteration a hyper-association is chosen from level 1 entity set. In subsequent iterations a hyper-association is chosen from the neighbors list which will be generated in step 4. The chosen hyper-association is passed to step 4 again to generate its neighbors. Steps 3 and 4 are performed repetitively.

4. Generate Neighbors

In this step all the neighbors i.e. sub neighbors and super neighbors are generated for hyper-association chosen in step 3. These neighbors are also hyper-associations. If the chosen hyper-association has size k then its sub neighbors will have size k - 1 and its super neighbors will have size k+1. Sub neighbors are generated by removing a single entity at a time from chosen hyper-association. For example if chosen hyper-association has entities “A, B” then its sub neighbors are entities A and B. Similarly super neighbors are generated by adding one entity from level 1 entity set at a time to the chosen hyper-association. For example if level 1 entity set contains entities: “A”, “B”, “C” and “D” and if chosen hyper-association has entities “A, B” then its super neighbors will have entities “A, B, C” and “A, B, D”.

When the new hyper-association is generated its VAT is calculated by taking the intersection of document-ids of its constituent entities. This will give the documents having the new hyper-association. For example if an hyper-association containing entities “A, B, D” was generated from “A, B” and “D”, we take intersection of documents in VAT of hyper-association “A, B” and documents in VAT of entity “D”. If “A, B” has a VAT of <D1,1>, <D2, 2>, <D3,5> and “D” has a VAT of <D1,3>, <D3,4>, <D4,4> then VAT of “A, B, D” would be <D1,min(1,3)>, <D3,min(4,5)> which is <D1,1>,<D3,4>. This set intersection was proposed in ECLAT. For every super neighbor and sub neighbor generated, the support of generated neighbor is calculated and if it is frequent it is added to the neighbors list of chosen entity. After generating all possible neighbors for a chosen hyper-association, that hyper-association is marked as visited and step 3 is repeated to choose a hyper-association randomly from neighbors list.

5. Frequent k Hyper-Association Extraction

This random walk comprising of step 3 and 4 is repeated for “n” times. “n” is a number predefined by the user. A condition is checked in step 3 to avoid choosing hyper-associations which were visited previously. This will help avoiding generating the neighbors for chosen hyper-association again. As only frequent hyper-associations in the neighbors are added to neighbors list we avoid choosing an infrequent hyper-association and generating its neighbors. This is important because all neighbors of an infrequent hyper-association can also be infrequent. After every random walk the hyper-associations in the neighbors list are added to the frequent hyper-associations list. At end of all “n” random walks, we have a list of frequent hyper-associations with support value of each hyper-association. Thus we have managed to generate a sample of frequent hyper-associations with significant performance improvement over exhaustive approaches.

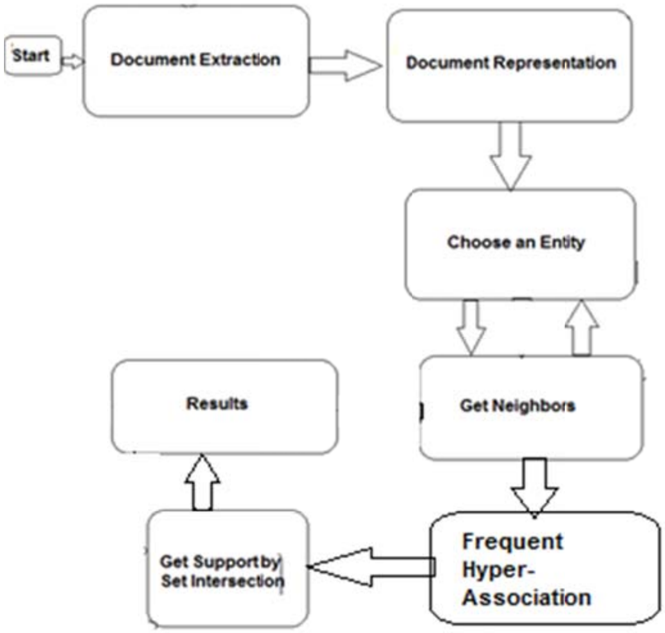

Figure 6 Output Space Sampling

3.2.3.1. Personalization Variant 1

In this approach user pre-selects the hyper-associations in level 1 entity set he/she is interested in. So during the Output space sampling process only these hyper-associations and its neighbors are considered. This means the random walk is performed only selected set of hyper-associations. This way we can avoid choosing all the hyper-associations and generating neighbors for them. Sampling is done on selective set of hyper-associations. Thus user can personalize the sampling space accordingly based on his/her choice. This approach will lead to an efficient algorithm as it is cheaper to run it on fewer hyper-associations than on all the hyper-associations.

3.2.3.2. Personalization Variant 2

In this approach user continuously provides feedback on the new hyper-associations generated. Thus sampling is done based on user feedback at each level. Instead of pre-selecting the hyper-associations at the start of random walk, user inputs his choice either YES or NO based on the new hyper-association generated by the algorithm. If user is interested in new hyper-association generated, he inputs YES. Based on this feedback algorithm generates all its frequent neighbors and presents one among them to the user. If user inputs NO to a hyper-association chosen from neighbors list, that hyper-association is never considered in random walk again and its neighbors are not generated. Then algorithm selects one of the hyper-association from previous neighbors list. In this approach a hyper-association can be chosen again if user is interested to explore more about its neighbors and find more frequent hyper-associations containing this hyper-association. This process is repeated for “n” times where “n” is defined by the user. In this approach we provide user with flexibility to choose hyper-association he/she is interested in at each level in random walk.

CHAPTER 4. RESULTS

4.1. Incremental Mining

We tested incremental mining approach to find the performance improvement over traditional text mining approaches on proteins data related to bone biology.

Table 4 Protein names

1	actin
2	alpha-actinin
3	architectural transcription factor
4	beta-catenin
5	ecm
6	hmg-motif
7	hmgb1
8	nmp4
9	p130cas
10	pth
11	pthr1
12	r-smad
13	receptor for advanced glycation end products
14	smad4
15	zyxin

Before performing the tests, a total of 120451 documents were downloaded from PubMed containing 15 proteins listed in table 4. This download process takes about 24 hrs. Then 2 runs of Incremental Miner are performed and were compared with a text mining approach without incremental mining. In the first run 9 proteins were used to extract associations among them, and then 6 more proteins were added in second run. Table 6 summarizes the time taken by each module of Incremental Mining. All the metrics are measured in seconds. Since documents were already downloaded prior to performing these tests, the time taken for download is not mentioned in table 6, but it will be a major bottleneck in approaches without Incremental Mining. The association strengths calculated for protein-protein combinations is shown in table 5.

Table 5 Association matrix for proteins

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1															
2	21796.7														
3	23.1	0													
4	1400.9	285.8	78.0												
5	1343.5	264.3	0	224.4											
6	208.6	27.9	145.5	15.8	10.5										
7	0	0	0	0	0	0									
8	89	0	588.7	190.1	22.6	52.8	0								
9	327.5	21.2	14.8	331.9	127.7	10.2	0	220.9							
10	1309.9	58.4	31.3	823.8	931.4	129.9	0	733.0	50.6						
11	19.8	0	0	3.4	12.6	0	0	0	0	10941.4					
12	6.1	0	0	0	0	603.2	0	0	0	0	0				
13	27.4	0	0	81.7	60.8	0	0	0	0	0	0	0			
14	197.4	3.1	0	990.2	337.7	0	0	0	0	235.1	0	0	3074.1		
15	1366.3	2503.4	8.1	54.2	92.5	5.6	0	151.6	377.2	26.5	0	0	0	0	

Table 6 Summary of time taken with and without Incremental Mining

	Incremental Miner	Incremental Miner	Text Miner	Text Miner
	First Run	Second Run	First Run	Second Run
Number of terms	9	15	9	15
Number of documents	42617	120451	42617	120451
Time taken for querying PubMed	0.10	0.10	0.10	0.10
Time taken for checking for new documents	3	11	NA	NA
Time taken for calculating weights	83	332	128	529
Time taken for calculating associations	1	1	0.62	0.89
Total time taken	87	361	130	541

4.2. Frequent Itemset Mining

We compared our Output space sampling approach with Apriori and ECLAT. All the three approaches were measured on three parameters:

1. Time taken for finding frequent co-occurring entities/hyper-associations
2. Precision
3. Recall

Precision and Recall are widely used metrics to determine the correctness of a pattern recognition algorithm. We used them to check how many similar and correct frequent hyper-associations were generated by Output space sampling approach. Precision is measure of exactness and Recall is measure of completeness. In association rule mining, Precision is defined as fraction of correct frequent hyper-associations to total extracted frequent hyper-associations. Recall is defined as fraction of correct frequent hyper-associations to total frequent hyper-associations for that data. Exhaustive approaches like Apriori, ECLAT generate all frequent hyper-associations. We have used protein data from bone biology to find frequent hyper-associations. The abstracts containing these proteins were downloaded from PubMed. The proteins chosen are mentioned in table 4.

4.2.1. Apriori

Apriori is a bottom-up breadth-first approach which generates all frequent hyper-associations. This approach was used on set of 15 proteins mentioned in table 4. The results for Apriori algorithm are shown in table 7.

Table 7 Performance of Apriori

Number of Entities	15
Number of Documents	118065
Number of Frequent Hyper-Associations	95
Total Time Taken	153 seconds
Precision	1.0
Recall	1.0

4.2.2. ECLAT

Similar to Apriori, ECLAT is an exhaustive approach which generates all the frequent hyper-associations. But the advantage is in its data representation format which enables efficient support counting. The results of ECLAT performance is mentioned in table 8.

Table 8 Performance of ECLAT

Number of Entities	15
Number of Documents	118065
Number of Frequent Hyper-Associations	95
Total Time Taken	0.828 seconds
Precision	1.0
Recall	1.0

4.2.3. Output Space Sampling

To test our Output space sampling approach we performed three cases studies with proteins data. In the first study we used all the 15 protein entities as input i.e. this is Output space sampling without personalization. Second and third studies were done with personalization variants mentioned in chapter 3. In second study user selects a set of entities of his/her interest from set of 15 entities, i.e. this is personalization variant 1. In third study, user continuously provides feedback to the system during all the random walks to let system know if he/she is interested in hyper-association generated by the system in that random walk i.e. this is personalization variant 2. Performance of each case study based on three metrics mentioned before is shown in tables 9, 10 and 11.

Case Study 1

In this case study we performed 50 random walks on the proteins data to find frequent hyper-associations. The time taken and number of frequent hyper-associations mentioned in table 9 are average values taken after running this approach for 10 times.

Table 9 Performance of Output Space Sampling without personalization

Number of Entities	15
Number of Documents	118065
Number of Frequent Hyper-Associations	75
Total Time Taken	0.657 seconds
Precision	1.0
Recall	0.789

Case Study 2

In this case study we performed same 50 random walks as above but user selected 9 entities of interest before the Output space sampling is done. The results are shown in table 10.

Table 10 Performance of personalization variant 1

Number of Entities	9
Number of Documents	118065

Number of Frequent Hyper-Associations	50
Total Time Taken	484 milliseconds
Precision	1.0
Recall	0.526

Case Study 3

In this case study user continuously provides feedback to system about the frequent hyper-associations generated on each random walk. User provides feedback for 50 times regarding the generated frequent hyper-associations. User selected 23 hyper-associations as interesting out of 50 hyper-associations generated by the system.

Table 11 Performance of personalization variant 2

Number of Entities	15
Number of Documents	118065
Number of Frequent Hyper-Associations	50
Number of Frequent Hyper-Associations Chosen By User	23
Total Time Taken	468 milliseconds
Precision	1.0
Recall	0.242

To test the correctness of Output space sampling approach, documents downloaded from PubMed were verified manually. Output space sampling extracted a hyper-association containing entities [actin, alpha-actinin, beta-catenin, zyxin] from documents with ids 10722635 and 14760006. Both these documents were verified manually to find the occurrence of all the entities of above hyper-association. Text containing the entities in hyper-association is highlighted in table 12.

Table 12 Text containing the entities of hyper-association

Document ID	Text in Document
10722635	cryptosporidium parvum induces host cell **actin** accumulation at the host-parasite interface.cryptosporidium parvum is an intracellular protozoan parasite that causes a severe diarrheal illness in humans and animals.The actin-binding protein **alpha-actinin** is also present in this plaque early in parasite development but is lost as the parasite matures. other actin-associated proteins, including vinculin, talin, and ezrin, are not present. we have found no evidence of tyrosine phosphorylation within this structure. molecules known to link actin filaments to membrane were also examined, including alpha-catenin, **beta-catenin**, plakoglobin, and **zyxin**, but none was identified at the host-parasite junction. thus, cryptosporidium induces rearrangement of the host cell cytoskeleton and incorporates host cell actin and alpha-actinin into a host-parasite junctional complex.
14760006	**zyxin**, axin, and wiskott-aldrich syndrome protein are adaptors that link the cadherin/catenin protein complex to the cytoskeleton at adherens junctions in the seminiferous epithelium of the rat testis.during spermatogenesis, the movement of germ cells across the seminiferous epithelium is associated with extensive junction restructuring. zyxin, axin, and wasp were shown to be structurally linked to the n-cadherin/beta-catenin/**alpha-actinin**/**actin** complex but not to the nectin-3/afadin or the beta 1-integrin-mediated protein complexes. interestingly, zyxin, axin, and wasp are also structurally linked to vimentin (an intermediate filament protein) and alpha-tubulin (the subunit of a microtubule), which suggests that they have a role (or roles) in the regulation of the dynamics of the desmosome-like junction and microtubule. these results illustrate that zyxin, axin, and wasp are adaptors in both ajs and intermediate filament-based desmosome-like junctions. this raises the possibility that classic cadherins are also associated with vimentin-based intermediate filaments via these adaptors in the testis. while virtually no n-cadherin was found to associate with vimentin in the seminiferous tubules, it did associate with vimentin when testis lysates were used. interestingly, about 5% of the e-cadherin associated with vimentin in isolated seminiferous tubules, and about 50% of the e-cadherin in the testis used vimentin as its attachment site. these data suggest that cadherins in the testis, unlike those in other epithelia, use different attachment sites to anchor the cadherin/catenin complex to the cytoskeleton. the levels of zyxin, axin, and wasp were also assessed during af-2364-mediated aj disruption of the testis, which illustrated a time-dependent protein reduction that was similar to the trends observed in nectin-3 and afadin but was the opposite of those observed for n-cadherin and **beta-catenin**, which were induced. collectively, these results illustrate that while these adaptors are structurally associated with the cadherin/catenin complex in the testis, they are regulated differently.

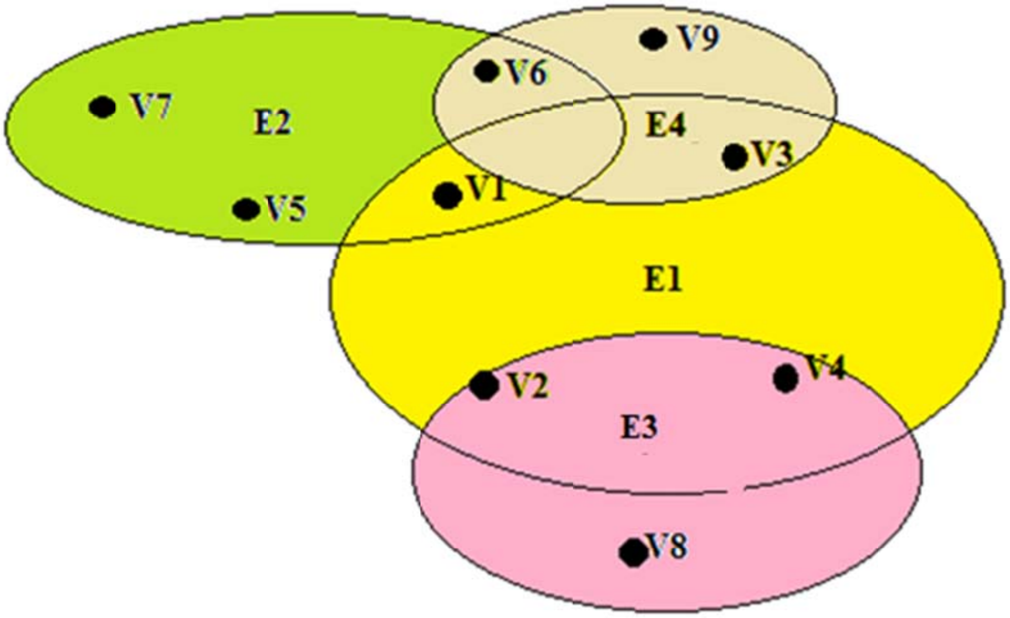

Figure 7 Sample hypergraph for proteins

Table 13 Sample hyper-associations extracted

Hyper-Edge	Entities in Hyper-Association
E1	actin, alpha-actinin, beta-catenin, zyxin
E2	actin, architectural transcription factor, nmp4, pth
E3	actin, beta-catenin, ecm
E4	nmp4, p130cas, zyxin

CHAPTER 5. CONCLUSION

In this thesis work we proposed a novel approach to find binary associations among various biological entities appearing in the medical literature. This approach has been proven to be better than traditional text mining approach using a concrete case study with protein data in field of bone biology. It has been found that incremental mining approach generates same associations as traditional methods but in very less time. These improvements can be even more drastic for large data sets. Hence this approach makes association extraction more practically useful. The associations extracted could then by verified by experiment. In future we hope to improve document extraction process by finding a mechanism to download all the documents together instead of sending request for each document. I believe this change will improve performance significantly.

In this thesis we also proposed a novel approach to personalized frequent hyper-associations extraction using the concept of Output space sampling. Finding frequent hyper-associations from literature can help researchers visualize new associations among entities previously unknown. This method provides cost-effective knowledge discovery with minimum human intervention. These novel hyper-associations can be verified later by experiment. We tested our approach on protein data from field of bone biology with abstracts downloaded from PubMed. The experimental results show that Output space sampling performs better than other previously known approaches. It is efficient in time and space as it doesn't involve candidate generation. This method is particularly useful when the data contains long frequently co-occurring entities. Further we provide a personalization approach where user can select the entities of his/her choice to find frequently co-occurring entities in text. We measured the performance of our approach on 3 parameters: Time taken to generate frequent hyper-associations, Precision and Recall. Output space sampling takes less time than ECLAT and Apriori to extract frequent hyper-associations. Our approach had a precision of 1.0 which implies all frequent hyper-associations were correct. Since our approach generates a sample of possible frequent hyper-associations, recall was not 1.0. There are several benefits in using Output space sampling approach. Few of them are:

1. Scalability: This approach is best to find "m" frequent hyper-associations. It is much cheaper to find these "m" samples as opposed to generating all possible combinations of these entities and selecting the frequent ones out of them.

2. Generic: This approach can be used to find any co-occurring entities. This approach takes the entities of interest and returns the frequently co-occurring entities. Thus the entities of interest can be changed without changing the algorithm.

3. Parallelizable: This approach can be simultaneously run on k machines to perform k random walks and extract frequent hyper-associations from each of those runs as there is no dependency among these walks. This leads to k-fold increase in speed and efficiency.

Hence Output space sampling offers a novel mechanism to find hyper-associations among the entities occurring in literature. The hyper-associations can be used to gain knowledge about the context and the entities. This knowledge can be presented to researchers in cognition-rich hypergraph to assist them in their research.

LIST OF REFERENCES

LIST OF REFERENCES

[1] Feldman, R. and Dagan, I. (1995) Knowledge discovery in textual databases (KDT). In proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD - 95), Montreal, Canada, August 20-21, AAAI Press, 112-117.

[2] Charniak, E. Introduction to artificial intelligence, page 2. Addison-Wesley, 1984.

[3] Poole, D. Mackworth, A. Goebel, R. (1998), Computational Intelligence: A Logical Approach, Oxford University Press, pp. 1, http://www.cs.ubc.ca/spider/poole/ci.html.

[4] Salton, G. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Reading, Addison-Wesley, 1989.

[5] Berge, C. Graphs and hypergraphs. Amsterdam, Netherlands: North-Holland Publishing; 1973.

[6] Piatetsky-Shapiro, G. (1991), Discovery, analysis, and presentation of strong rules, in Piatetsky-Shapiro, G., and Frawley, eds W. J., ‘Knowledge Discovery in Databases’, AAAI/MIT Press, Cambridge, MA.

[7] Hastings, W.K. (1970). "Monte Carlo Sampling Methods Using Markov Chains and Their Applications". Biometrika 57 (1): 97-109. [doi:10.1093/biomet/57.1.97].

[8] PubMed is a free database accessing the MEDLINE database of citations, abstracts and some full text articles on life sciences and biomedical topics [Internet]. Available online at: http://www.ncbi.nlm.nih.gov/pubmed.Last Accessed on 03/08/2011.

[9] Vaka, H.G.G., Mukhopadyay, S. Knowledge Extraction and Extrapolation Using Ancient and Modern Biomedical Literature. Accepted for 2008 IEEE BioCom Workshop 2009, conjunction with AINA 2009.

[10] Ayurveda [Internet], Available on: http://en.wikipedia.org/wiki/ayurveda. Last Accessed on 03/08/2011.

[11] Nobata, C., Collier, N., and Tsujii, J. (1999). Automatic term identification and classification in biology texts. In Proceedings of the 5th natural language processing pacific rim symposium (pp. 369-374).

[12] MedlinePlus [Internet]. Bethesda (MD): National Library of Medicine (US); [updated 2005 Aug 12; cited 2010 March 11]. Available from: http://medlineplus.gov/. Last Accessed on 03/08/2011.

[13] Kostoff, R.N., del Río, J.A., Humenik, J.A., García, E.O. and Ramírez, A. M. (2001), Citation mining: Integrating text mining and bibliometrics for research user profiling. Journal of the American Society for Information Science and Technology, 52: 1148-1156. [doi: 10.1002/asi.1181].

[14] Smalheiser, N.R. Predicting emerging technologies with the aid of text-based data mining: the micro approach, Technovation, Volume 21, Issue 10, October 2001, Pages 689-693, ISSN 0166-4972, [doi: 10.1016/S0166-4972(01)00048-7].

[15] Narayanasamy, V., Mukhopadhyay, S., Palakal, M., and Potter, D. (2004). TransMiner: Mining Transitive Associations among Biological Objects from Text. Journal of Biomedical Sciences, 11(6): 864-873.

[16] Donaldson, I., Martin, J., de Bruijn, B., Wolting, C., Lay, V., Tuekam, B., Zhang, S., Baskin, B., Bader, G.D., Michalickova, K., Pawson, T. and Hogue, C.W.V. (2003), PreBIND and Textomy – mining the biomedical literature for protein-protein interactions using a support vector machine. BMC Bioinformatics.

[17] Rindflesch, T.C., Tanabe, L., Weinstein, J.N., and Hunter, L. EDGAR: extraction of drugs, genes and relations from the biomedical literature. Pac Symp Biocomput. 2000, 517-28.

[18] Srinivasan, P. (2004), Text mining: Generating hypotheses from MEDLINE. Journal of the American Society for Information Science and Technology, 55: 396-413. [doi: 10.1002/asi.10389].

[19] Swanson, D.R. and Smalheiser, N.R. (1997). An interactive system for finding complementary literatures: a stimulus to scientific discovery. Artificial Intelligence 91: 183-203.

[20] Nenadić, G., Ananiadou, S. (2006), Mining semantically related terms from biomedical literature. ACM Transactions on Asian Language Information Processing Volume 5 Issue 1, March 2006.

[21] Yeganova, L., Smith, L., Wilbur, W. J. Identification of related gene/protein names based on an HMM of name variations, Computational Biology and Chemistry, Volume 28, Issue 2, April 2004, Pages 97-107, ISSN 1476-9271,[doi:10.1016/j.compbiolchem.2003.12.003].

[22] Mukhopadhyay, S., Palakal, M., Maddu, K. Multi-way Association Extraction from Biological Text Documents Using Hypergraphs, Bioinformatics and Biomedicine, 2008. BIBM '08. IEEE International Conference, 257-262, 2008.

[23] Palakal M., Stephens M., Mukhopadyay S., Raje R. Identification of biological relationships from text documents using efficient computational methods, Journal of Bioinformatics and Computational Biology, Volumes 1, 2, 2003, 307-34.

[24] Agrawal, R. and Srikant, R. Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pages 487-499, Santiago, Chile, September 1994.

[25] Han, J., Pei, J., Yin, Y. and Mao, R. Mining frequent patterns without candidate generation. Data Mining and Knowledge Discovery 8:53-87, 2004.

[26] Webb, G.I. (1995). OPUS: An Efficient Admissible Algorithm for Unordered Search. Journal of Artificial Intelligence Research 3.

[27] OneR, Ross, Peter. [Internet] Available online: http://www.soc.napier.ac.uk/~peter/vldb/dm/node8.html. Last Accessed on 03/08/2011.

[28] Zaki, M.J. Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering, 12(3):372-390, May/June 2000.

[29] Rauch, J. Logical calculi for knowledge discovery in databases. Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery, Springer, 1997, pages. 47-57.

[30] Gouda, K. and Zaki, M.J. Efficiently mining maximal frequent itemsets. In 1st IEEE International Conference on Data Mining, November 2001.

[31] Hasan, M. and Zaki, M. "Output Space Sampling for Graph Patterns," Proc. Very Large Databases Conf., 2009.

[32] Masseglia, F., Poncelet, P. and Teisserie, T. Incremental mining of sequential patterns in large databases. [doi:10.1016/S0169-023X(02)00209-4] .

[33] El-Sayed, M., Ruiz, C., Rundensteiner, E.A. FS-Miner: efficient and incremental mining of frequent sequence patterns in web logs, [doi:10.1145/1031453.1031477]