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 Shruti Dilip Waranashiwar

Entitled
Interactive Pattern Mining of Neuroscience Data

For the degree of Master of Science

Is approved by the final examining committee:

Dr. Snehasis Mukhopadhyay
Chair

Dr. Arjan Durresi

Dr. 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 05/30/2013
Head of the Graduate Program Date

INTERACTIVE PATTERN MINING OF NEUROSCIENCE DATA

A Thesis

Submitted to the Faculty

of

Purdue University

by

Shruti Dilip Waranashiwar

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

August 2013

Purdue University

Indianapolis, Indiana

To,

My parents, husband and son.

ACKNOWLEDGEMENTS

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. Arjan Durresi and Dr. Yuni Xia for agreeing to be a part of my thesis committee. I thank Dr. Mohammad Al Hasan and his student Mansurul Bhuiyan for providing me guidance during various stages of my thesis work. I am also very thankful to Dr. Christopher Lapish from Department of Psychology for providing inputs from neuroscience perspective.

I would like to thank my family for their unconditional love and support. I also want to thank all my friends and well-wishers for their good wishes and support.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Waranashiwar, Shruti Dilip. M.S., Purdue University, August 2013. Interactive Pattern Mining of Neuroscience Data. Major Professor: Snehasis Mukhopadhyay.

Text Mining is a process of extraction of knowledge from unstructured text documents. We have huge volumes of text documents in digital form. It is impossible to manually extract knowledge from these vast texts. Hence, text mining is used to find useful information from text through the identification and exploration of interesting patterns. The objective of this thesis in text mining area is to find compact but high quality frequent patterns from text documents related to neuroscience field. We try to prove that interactive sampling algorithm is efficient in terms of time when compared with exhaustive methods like FP Growth using RapidMiner tool. Instead of mining all frequent patterns, all of which may not be interesting to user, interactive method to mine only desired and interesting patterns is far better approach in terms of utilization of resources. This is especially observed with large number of keywords. In interactive patterns mining, a user gives feedback on whether a pattern is interesting or not. Using Markov Chain Monte Carlo (MCMC) sampling method, frequent patterns are generated in an interactive way. Thesis discusses extraction of patterns between the keywords related to some of the common disorders in neuroscience in an interactive way. PubMed database and keywords related to schizophrenia and alcoholism are used as inputs. This thesis reveals many associations between the different terms, which are otherwise difficult to understand by reading articles or journals manually. Graphviz tool is used to visualize associations.

CHAPTER 1. INTRODUCTION

1.1 Text Mining

Nowadays, huge volumes of research literatures are available online. Pubmed, Medline are few of many medical literature databases. This abundance of data sources is full of information and knowledge. But it is not possible to extract all knowledge from text manually. Manual method may result in overlooking some important information. There is also possibility of misinterpretation. Hence, text mining, an automated approach is solution for all such problems.

In text mining, a user interacts with a document collection over time using text mining tool. It is a knowledge-intensive process. Here data sources are unstructured textual data in the documents. Text mining requires preprocessing of test data. Preprocessing operations include identification and extraction of representative features for text documents. It results in transformation of unstructured data stored in document collection into a more explicitly structured intermediate format.

Concepts of text mining cover areas of information retrieval (IR), information extraction (IE), natural language processing (NLP) [1] and artificial intelligence (AI). IR is responsible for storing, searching and retrieving information like stemming etc. While IE is concerned with the extraction of semantic information from text ex. named entity recognition [2]. NLP covers tasks like parts-of speech tagging, parsing etc. Text mining also includes concepts of learning (supervised and non-supervised) from AI. NLP and AI have wide range of applications including human-computer interaction, medical research, stock trading, and robot control.

In this thesis, we try to extract novel and interesting associations among keywords appearing in PubMed abstracts using text mining approaches. These associations should later be validated by experiments. Main steps considered in extraction of associations by text mining are: document extraction, document parsing, and document representation, weight computation for entities, association matrix computation for associations among the entities and frequent pattern computation. Generally, well known TF-IDF algorithm [3] is used for assigning scores to the keywords associations. Details about frequent pattern computation are discussed in later chapters.

Text mining approach extracts binary and transitive associations among keywords. Binary associations are direct associations between two keywords. Transitive associations are extracted from existing known associations using the transitivity property, which results in novel and currently unknown associations. Transitive associations are treated as plausible hypotheses and can be subjected to further validations. In simple terms, transitive association is - If A is related to B and B is related to C, then A may be related to C. Transitive associations help to understand the context of association. For example, “Keyword A interacts with keyword B in domain C under influence of keyword D”.

1.2 Schizophrenia and Alcoholism – Neuroscience Perspective

Neuroscience is the scientific study of the nervous system [17]. It is not only a branch in biology anymore; it is broadened to include other fields such as psychology, philosophy and computer science. Nervous system is the body's major communication system. Neuroscience focuses on various aspects of nervous system. How the nervous system works, develops, malfunctions and how it can be repaired are some of the topics of research. There has been lot of progress in neuroscience research because of developments in computers, brain imaging, genetics and genomics.

Alcoholism and schizophrenia commonly co-occur. People with schizophrenia are much more likely to have an alcohol abuse problem than the general population [16]. Neuroscience has made a number of advances in understanding neurobiological changes

which occur with repeated heavy use of a substance. Substance dependence is considered as a disorder that involves the motivational systems of the brain. The disorder of the brain results in complex behavioral symptoms. New technologies provide a means to visualize and measure changes in brain function that occur with short-term or long-term abuse of substance [8].

Schizophrenia is a chronic and severe psychiatric disorder. The person with schizophrenia shows symptoms of persistent delusions, hallucinations, disorganized speech, and disorganized behavior. It is also characterized by deficit of emotional expression or a lack of motivation or initiative. They may hear voices (hallucinations) other people don't hear. They fear other people are controlling their thoughts. Lack of responsiveness and disorganized thinking and speech are also often observed. Difficulty in working and long-term memory loss, attention makes them hard to hold a job and take care of themselves. They become dependent.

Researchers think that schizophrenia is caused by several factors. Genes and environment, brain chemistry and structure may be major factors. If schizophrenia is present in family members, risk of having schizophrenia increases. Scientist believes several genes are associated with this disorder but no single gene is responsible. It may also result when certain gene that makes important brain chemicals malfunctions. Many environmental factors also contribute to it. Imbalance in complex, interrelated chemical reactions of the brain involving the neurotransmitters dopamine and glutamate, are also key factors in schizophrenia.

Yet, no single organic cause of schizophrenia has been found. Current research is focused on neurobiology and psychology for the treatment of this disorder. Major treatment for schizophrenia is antipsychotic medication, which primarily suppresses dopamine and sometimes serotonin receptor activity [8].

Schizophrenia is frequently shows comorbidity with medical illnesses, mental retardation, and substance abuse. Most common is the substance use disorder. Nicotine and alcohol are the most common substance of abuse. Scientists have learned a lot about schizophrenia and alcoholism, but more research is needed to explain if there is any relationship between alcoholism and schizophrenia.

Alcoholism is alcohol abuse in terms of psychiatry. Alcoholism poses threats to drinker's personal relationships, social standing and health. Alcohol is consumed throughout the world for various purposes like social, recreational. But because of genetic variations in metabolic enzymes, effects of alcohol on individual are different. Thus some people become addict and others are not.

Alcohol, i.e. ethanol consumption results in sedative effects and impairment of memory during periods of intoxication. Chronic consumption of alcohol can alter some of the brain systems and structures. These alterations are correlated with impairments in cognitive processes. Prolonged use of alcohol may result in lack of judgment and difficulty in decision-making and emotional disturbance. Alcoholism causes damage to brain functions and also affects psychological health.

To treat alcoholism, these cognitive impairments must be targeted. Alcohol abuse can be treated with detoxification, psychological and medical treatments. Treatment of alcoholism includes acamprosate, a synthetic drug which acts centrally and appears to restore normal activity of neurons [8]. Disulfiram is known as 'deterrent' is also used to make indigestion to alcohol unpleasant and helps to treat alcoholism. Tolerance to the effects of sedatives develops rapidly and doses needs to be increased. Upon withdrawal of sedatives, anxiety, restlessness and insomnia are observed. That leads to difficulty in completely treating this disorder.

Psychiatric disorders have seen to co-occur with alcoholism. Psychosis, confusion and organic brain syndrome may be caused by excessive use of alcohol. Schizophrenia also shows similar symptoms. Hence, there is a need to find if there is some association between keywords related to these two disorders – schizophrenia and alcoholism and what the strength of these associations between different keywords is.

There is a lot research going on to study comorbidity between various psychiatric disorders. Mental disorders and substance use disorders have shared neurobiological and behavioral abnormalities. So, this similarity suggests that these disorders are linked. But the main of cause of comorbidity is still not clear. It is still not sure that which factors leads to this comorbidity. Precise association is not known yet but more research will help to clarify the treatment of schizophrenia and alcoholism.

Several biological, psychological, and socio-environmental factors are contributing for the co-occurrence of schizophrenia and alcoholism. Homelessness, poverty, boredom and mental illness are also associated with comorbidity of schizophrenia and alcoholism. There are three biological factors. People with schizophrenia use drugs to self-medicate in an attempt to alleviate the symptoms of schizophrenia or side effects of treatment drugs of schizophrenia. Brain abnormalities in schizophrenia help reinforcing effects of alcohol abuse. Use of schizophrenia syndrome produces impaired thinking and social judgment and poor impulse control.

People with substance abuse don't follow treatments plan, so it makes treatment less effective. Mostly, research in field of neuroscience is performed on individual disease states. So it leads to dearth of information regarding how these diseases may interact with each other. Abnormalities related to schizophrenia and effects of alcohol abuse may interact with some of the brain systems. To study comorbidity between these two disorders, two set of terms are used, one for alcoholism and other for schizophrenia. These terms related to the specific neurological disorders are selected with the help of neuroscience expert. 18 terms are used for alcoholism and 20 for schizophrenia.

Here we try to give brief idea about keywords that we have considered for thesis.

Gamma amino butyric acid (GABA-A) receptors are sensitive to ethanol in distinct brain regions and thus contribute to severe effects of ethanol, ethanol tolerance and dependence [8]. Addiction is a continued use of mood altering substance [9] and it includes drug abuse, impaired control of substances. Alcoholism is already discussed in previous pages.

Table 1.1 Alcoholism Terms

Sr. No.	Alcoholism Terms
1	Gamma amino butyric acid
2	Alcohol preferring rat
3	Addiction
4	Alcoholism
5	Dopamine
6	Glutamate
7	Catechol-o-methyl-transferase
8	Prefrontal cortex
9	Anterior cingulate cortex
10	Prelimbic cortex
11	Ventral tegmental area
12	Nucleus accumbens
13	Orbitofrontal cortex
14	Cognition
15	Alcoholic hallucinosis
16	Nmda
17	Serotonin
18	Disulfiram

Dopamine is a neurotransmitter that is derived from amino acid tyrosine. It plays important role in movement, learning and motivation. Glutamate is an excitatory amino acid neurotransmitter found throughout the brain. It is important for learning and plays an essential role in the hippocampus. Catechol-o-methyl-transferase (COMT) inactivates catecholamine and catechol drugs. COMT activity could partially affect the appearance of delirium tremens in these individuals. Prefrontal cortex contributes to the malfunction of nucleus accumbency system. Cognition is a group of mental processes that includes attention, memory, producing and understanding language and decision making.

Alcoholic hallucinosis is a complication of alcohol withdrawal in alcoholics. Nmda is an amino acid and is related to control of memory function. Serotonin is a monoamine neurotransmitter and contributes to feeling of well-being and happiness. Disulfiram is a drug designed for the treatment of alcoholism. Ketamine is a drug used in the treatment of depression in patients. Hypofrontality contributes to the cognitive deficits associated with schizophrenia.

Table 1.2 Schizophrenia Terms

Sr. No.	Schizophrenia Terms
1	Gamma amino butyric acid
2	Gamma oscillation
3	Neonatal ventral hippocampal lesion
4	Schizophrenia
5	Dopamine
6	Glutamate
7	Catechol-o-methyl-transferase
8	Prefrontal cortex
9	Anterior cingulate cortex
10	Prelimbic cortex
11	Ventral tegmental area
12	Nucleus accumbens
13	Orbitofrontal cortex
14	Cognition
15	Nmda
16	Nmda hypofunction
17	Hypofrontality
18	Ketamine
19	Serotonin
20	Sensitization

PubMed is a free database of references and abstracts on life sciences and biomedical topics. PubMed has lot of text documents related to neuroscience field. Advancements in the field of biology, neuroscience, and psychology lead to enormous increase in literature. However despite this large knowledge base, we are not able to find therapeutic treatment for disorders like schizophrenia and alcoholism. This work may help in medical research

to find treatment for schizophrenia and alcoholism. We can use text mining techniques to predict novel relationships that may strongly contribute in treatment of such disorders. Text mining can provide an automated approach to mine this vast text and this thesis tries to find relationship between these two disorders. Association between keywords is assigned a value to determine strength and relevance of association. Also, in this thesis we use text mining to go beyond the list of supplied keywords and retrieve new terms from the text documents that are not supplied by researcher. Thus, text mining is a powerful tool to automatically extract associations from dynamic information sources, e.g., PubMed.

1.3 Pattern Mining

Traditional text mining includes retrieving set of documents by querying set of keywords of interest. These set of documents are subjected to text mining tools to extract associations based on co-occurrence of keywords. These associations are assigned scores and user-defined threshold is applied to filter out strong binary associations. Binary associations are further used to retrieve transitive associations. This method results in extracting all the binary associations, some of which may not be that useful to user. Also, as the size of data and number of keywords increases, this approach becomes less efficient. So, in this thesis we propose interactive pattern mining approach which does text mining in interactive way based on feedback from user and associations retrieved from preceding iterations.

The objective of this thesis is to prove that an interactive pattern mining (IPM) approach to extract associations among keywords is far efficient than traditional text mining methods. In order to prove this, we compared traditional text mining approach like FP-Growth algorithm with interactive sampling algorithm, which uses IPM approach. We tried to find all the associations using text mining tool - RapidMiner and compared time required to find these associations.

1.4 RapidMiner and IPM

RapidMiner is an open-source text mining tool. Here we use RapidMiner 5.2.008. RapidMiner provides data mining and machine learning procedures including: data loading and transformation (extract, transform, load, a.k.a. ETL), data preprocessing and visualization, modeling, evaluation, and deployment [4]. In this thesis, we use RapidMiner processes – retrieve, preprocessing, FP-Growth and association rule generator.

Interactive sampling [5] algorithm propose a novel approach called output space sampling which is a random walk algorithm of type Metropolis-Hastings based on Markov Chain Monte Carlo class of algorithms. The textual data used is neuroscience data related to schizophrenia and alcoholism. Database is PubMed which is an online repository for medical literature.

In IPM, user continuously provides feedback to each pattern chosen by system during random walk. Next frequent pattern is selected based on user's feedback to earlier pattern. Each pattern is send to user for his feedback. Thus, at the end of algorithm result includes only those frequent patterns that user is interested in.

This thesis is divided into 2 parts: text mining using RapidMiner and frequent item set mining using IPM. Preprocessing of input text data is done using MySQL database.

CHAPTER 2. BACKGROUND

This thesis draws motivation from paper – “Interactive pattern mining on hidden data: a sampling-based solution” by Mohammad Al Hasan, Snehasis Mukhopadhyay, Mansurul Buiyan [5] which describes mechanism to find interactive frequent patterns using interactive sampling algorithm. In this research, authors propose a solution that is based on Markov Chain Monte Carlo (MCMC) sampling of frequent patterns. Instead of returning all the frequent patterns, the proposed paradigm returns a small set of randomly selected patterns. It also allows interactive sampling, so that the sampled patterns can fulfill the user’s requirement effectively. This thesis is an extension to work done in [5] in terms of comparison of IPM with RapidMiner traditional approach. Thesis uses text data set and the text data is from neuroscience domain related to schizophrenia and alcoholism.

Paper "Text mining for neuroscience: a co-morbidity case study" by Christopher C. Lapish, Naveen Tirupattur, Snehasis Mukhopadhyay [6] covers how text mining techniques can be used to find associations between two neuroscience disorders - schizophrenia and alcoholism. It gives automated approach to study comorbidity between these two diseases. It also covers concepts of weight matrix computation using tf.idf, association matrix computation, adjacency matrix computation, transitive closure computation.

Paper “Identification of biological relationships from text documents” by Mathew Palakal, Snehasis Mukhopadhyay, and Matthew Stephens [12] presents a novel approach to extract relationships between multiple biological objects that are present in a text document. The approach involves object identification, reference resolution, ontology,

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.

There is significant research going on in the area of frequent pattern mining. “Frequent pattern mining: current status and future directions” is a paper in this area by Jiawei Han · Hong Cheng, Dong Xin, [7]. Here authors provide a brief overview of the current status of frequent pattern mining and discuss a few promising research directions. Authors mainly discusses following : (1) efficient and scalable methods for mining frequent patterns , (2) mining interesting frequent patterns, (3) impact to data analysis and mining applications, (4) applications of frequent patterns. Basic mining methodologies: Apriori, FP-growth and Eclat are discussed in this paper. Author also describes constraint-based mining, mining compressed or approximate patterns, frequent pattern-based classification, frequent pattern-based cluster analysis, frequent pattern analysis versus cube computation, gradient mining and discriminant analysis and applications like spatiotemporal and multimedia data mining, mining data streams, software bug mining and system caching, indexing and similarity search of complex structured data.

In paper “Information retrieval by semantic analysis and visualization of the concept space of d-lib magazine” by Junliang Zhang, Javed Mostafa, Himansu Tripathy [18], authors developed automatic techniques for term set extraction, association generation and visualization of concept spaces to aid retrieval from online document collections. Based on automatic concept discovery and clustering, the interface they developed visually depicts the generated concepts and their semantic relationships by using a spring embedding graph. The visualization provides the user a clear and attractive overview of what is available in a document collection. Authors are planning further improvement of the visualization algorithm implemented in this study.

In paper “Interactive constrained association rule mining” by Bart Goethals, Jan Van den Bussche [29], authors investigates ways to support interactive mining sessions, in the

setting of association rule mining. In such sessions, users specify conditions (queries) on the associations to be generated. Their approach is a combination of the integration of querying conditions inside the mining phase, and the incremental querying of already generated associations. Authors present several concrete algorithms and compare their performance. This paper also describes apriori algorithm, conjunctive constraints, Boolean queries, interactive mining approaches like integrated querying/post-processing and incremental querying.

Book, “Automatic text processing” by Gerard Salton [10] explains concept of text mining in detail. It covers topics of text analysis and language processing- language analysis and understanding, automatic indexing and advanced information retrieval models etc.

Book, “Text mining and its applications to intelligence, CRM and knowledge management” by A. Zanasi [11] presents advances in theory of applications management information. It covers concepts of data mining, text mining, intelligent agents, information retrieval, data warehousing, text processing and information retrieval, information extraction, text clustering, text categorization, summarization and visualization, applications of text mining.

I studied the dissertation “Continuous analysis of internet text by artificial neural network” submitted by Peter Jorgensen [32]. This dissertation details about concepts in text-mining and artificial neural network. This study explores the use of an interactive activation with competition (IAC) artificial neural network to aid in processing text to find relationships.

CHAPTER 3. METHODOLOGY

This thesis is broadly divided into three parts:

1) Text Document Preprocessing
2) Frequent Pattern Mining by FP Growth Algorithm
3) Interactive Sampling Algorithm

Following chapter contains detailed descriptions of all work done as part of this thesis.

3.1 Text Document Preprocessing

In this thesis we find frequent patterns from text documents related to neuroscience field using RapidMiner - FP Growth algorithm and IPM. We need to do preprocessing of text documents before we input it to algorithms.

This process has 4 major tasks:

1. Document Extraction
2. Frequent Keywords Extraction
3. Document Representation
    3.1 Document Representation for RapidMiner
    3.2 Document Representation for IPM

Abstracts containing the keywords of interest are downloaded by querying PubMed database.100 frequent keywords are extracted from this abstracts. Again new abstracts are retrieved for 100 keywords. This text data is stored in database for further processing. Finally text data is represented in suitable format to feed as input file.

3.1.1 Document Extraction

From 18 alcoholism keywords and 20 schizophrenia keywords, 25 distinct keywords are selected. Duplicates are not considered. Final list of 25 keywords is as below. Pubmed database is queried with this set of 25 keywords. OR query is used to extract abstracts. The query returns 285,060 numbers of abstracts.

Table 3.1 List of 25 Keywords

Sr. No.	Keywords
1	Alcohol preferring rat
2	Addiction
3	Alcoholic hallucinosis
4	Alcoholism
5	Anterior cingulate cortex
6	Catechol-o-methyl-transferase
7	Cognition
8	Disulfiram
9	Dopamine
10	Gamma amino butyric acid
11	Gamma oscillation
12	Glutamate
13	Hypofrontality
14	Ketamine
15	Neonatal ventral hippocampal lesion
16	Nmda
17	Nmda hypofunction
18	Nucleus accumbens
19	Orbitofrontal cortex
20	Prefrontal cortex
21	Prelimbic cortex
22	Schizophrenia
23	Sensitization
24	Serotonin
25	Ventral tegmental area

Table 3.2 Pubmed Query with 25 Keywords

((((((((((((((((((((((((gamma amino butyric acid[Title/Abstract]) OR gamma oscillation[Title/Abstract]) OR neonatal ventral hippocampal lesion[Title/Abstract]) OR Schizophrenia[Title/Abstract]) OR dopamine[Title/Abstract]) OR Glutamate[Title/Abstract]) OR Catechol-o-methyl-transferase[Title/Abstract]) OR prefrontal cortex[Title/Abstract]) OR anterior cingulate cortex[Title/Abstract]) OR prelimbic cortex[Title/Abstract]) OR ventral tegmental area[Title/Abstract]) OR nucleus accumbens[Title/Abstract]) OR Orbitofrontal cortex[Title/Abstract]) OR cognition[Title/Abstract]) OR Nmda[Title/Abstract]) OR Nmda hypofunction[Title/Abstract]) OR Hypofrontality[Title/Abstract]) OR Ketamine[Title/Abstract]) OR Serotonin[Title/Abstract]) OR Sensitization[Title/Abstract]) OR Alcohol preferring rat[Title/Abstract]) OR addiction[Title/Abstract]) OR alcoholism[Title/Abstract]) OR alcoholic hallucinosis[Title/Abstract]) OR Disulfiram[Title/Abstract]

We put following filters to choose only relevant queries -

1) Query Builder with title/abstract and query type ‘OR’,

2) Text availability - Abstract available,

3) Languages – English,

4) Publication dates - 10 Years.

The set of abstracts is downloaded in form of XML format. This XML data is parsed to extract abstract text from file. The text for each abstract is stored as new separate line. This input document is then input to RapidMiner for preprocessing.

3.1.2 Frequent Keywords Extraction

The input file with 285,060 abstracts is fed to RapidMiner text preprocessing tools. List of frequent keywords is generated based on number of occurrences and document occurrences. Common and insignificant words are filtered out and only significant top 100 keywords are selected from this list manually.

Keywords count of 60 is again queried with 'OR' to download new set of abstracts from the PubMed. Same filters that we use for 25 keywords are used here. Again text documents in the format of XML are parsed for abstracts and stored as separate row in file.

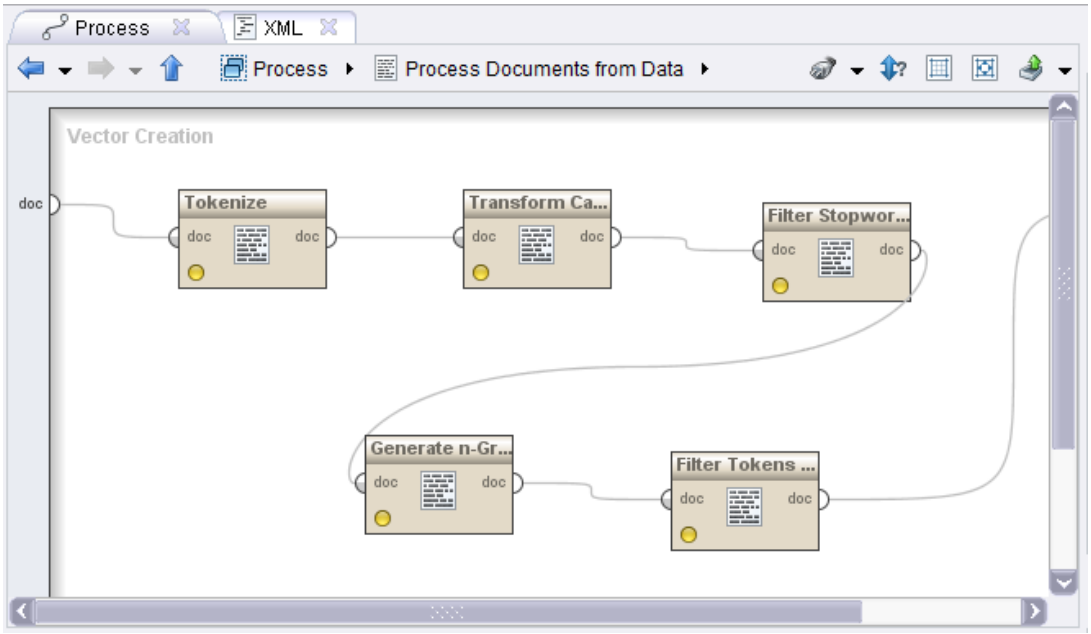

Figure 3.1 Preprocessing Operations in RapidMiner

.

Tokenize, transform cases, filter stop words, generate n-grams, filter tokens by length are the operators that we use to generate list of frequent keywords. Tokenize is use to tokenize the document by splitting the document into a sequence of tokens. Transform cases operator is applied after tokenize operator. Transform cases is used to convert all the characters in document to lower case. Filter stop words operator is used to remove all the tokens that equal stop words from the built-in stop words list. Generate n-grams creates term n-grams of tokens in a document. Filter tokens operator filter tokens based on their length. The database is created to store such large volume of text data. Abstracts extracted from database are stored in database MySQL. Database table to store the abstract text is created. Each row in table stores one abstract. This abstracts are CLOB data type. One column for each keyword is also added in table. This database table is further modified to present data in required form.

3.1.3 Document Representation

3.1.3.1 Document Representation for RapidMiner

Document representation required as input for RapidMiner tool and IPM is different.
For RapidMiner matrix of abstracts is created based on occurrence of keyword in abstract. Here documents are represented as rows while keywords as columns. SQL queries are written in MySQL to check the occurrences of keywords in abstracts. If particular keyword is present in abstract then value in matrix becomes 1. If the keyword is not present value of keyword for that particular abstract becomes zero. In such way, binary matrix of 1 and 0 is constructed for complete set of abstracts and keywords. The document matrix constructed is then written to a file and in RapidMiner, this file is read for further processing. So, we have the document representation of all abstracts and keywords appearing in those abstracts. The document format is shown in table 3.3
Here, keyword 1 in present in abstract no. 1. Keyword 2 in not present in abstract no. 1.

Table 3.3 Document Representation for RapidMiner

Abstract	Keyword 1	Keyword 2	Keyword 3	-----	Keyword 25
1	1	0	0	------	0
2	1	0	1	------	1
3	1	1	0	------	0

3.1.3.2 Document Representation for IPM

Document representation required as input for IPM system is quite different than RapidMiner. For IPM system, matrix of abstracts is created based on presence of keywords in abstract. Here documents are represented as rows while column data is created depending on what keywords are present. Keywords are mapped to numbers. SQL queries are written in MySQL to check the occurrences of keywords in abstracts. If particular keyword is present in abstract then number of that keyword is stored. If the keyword is not present number of keyword is not stored. In such way, mapping is done

for all the keywords and abstracts. The document matrix constructed is then written to a file and this file is fed as input to interactive sampling algorithm. So, we have the document representation of all abstracts and keywords appearing in those abstracts. The document format is shown in table 3.4. In abstract no 1, keywords mapped to number 2 4 6 10 45 are present and total count of keywords present are 5. Below is the sample example.

Table 3.4 Document Representation for IPM

Abstract	Keywords present	Total Keywords
1	2 4 6 10 45	5
2	5 6 23 56	4
3	4 8 12	3

3.2 Frequent Pattern Mining by FP Growth Algorithm

RapidMiner is the world-leading open-source system for data mining and text mining. It is available as a stand-alone application for data analysis and as a data mining engine for the integration into own products. It provides data mining and machine learning procedures including: data loading and transformation (extract, transform and load), data preprocessing and visualization, modeling, evaluation, and deployment [13].

Following are the main processes for frequent pattern mining by RapidMiner

1) Read Input Data
2) Process Documents from Data
    2.1Tokenize
    2.2Transform Cases
    2.3 Filter Stop Words
    2.4 Generate N-Grams (Terms)
    2.5 Filter Tokens by Length
3) FP-Growth
4) Association Rules

5) Drawbacks of Exhaustive Frequent Pattern Mining by FP Growth

RapidMiner has powerful and intuitive graphical user interface. It has repositories for process, data and metadata handling. It is flexible with hundreds of data loading, data transformation, data modeling, and data visualization methods.

3.2.1 Read Input Data

This operator reads file in csv format. Output is fed to next process of “Process documents from data”. Below is the description of the process.

Here all values of an example are written into one line and separated by a constant separator. The separator might be specified in the column separators parameter. The default will split the line on each comma, semicolon and blank. Arbitrary regular expressions are usable as separator. Empty values and the question mark will be read as missing values. You can quote the values (including the column separators) with a double quote ("). You can escape the quoting character with a backslash, i.e. \". The first line is used for the attribute names as default, controlled by the use first row as attribute names parameter. This operator tries to determine an appropriate type of the attributes by reading the first few lines and checking the occurring values. If all values are integers, the attribute will become integer, if real numbers occur, it will be of type real. Columns containing values which can't be interpreted as numbers will be nominal, as long as they don't match the date and time pattern of the date format parameter. If they do, this column of the csv file will be automatically parsed as date and the according attribute will be of type date [4].

3.2.2 Process Documents from Data

This operator generates word vectors from string attributes. Input is example set and output is word list. Following are the parameters of this process [4]

a) Create word vector: If checked, the tokens of a document will be used to generate a vector numerically representing the document. Range: Boolean; default: true

b) Vector creation: Select the schema for creating the word vector. Range: tf-idf, term frequency, term occurrences, binary term occurrences; default: tf-idf

c) Add meta information: If checked, available meta information of the text like filename, date is added as attribute. Range: Boolean; default: true

d) Keep text: If checked, the input text will be stored as a special string attribute with the role text. Range: Boolean; default: false

e) Prune method: Specifies if too frequent or too infrequent words should be ignored for word list building and how the frequencies are specified. Range: none, percent, absolute, by ranking; default: none

f) Prune below percent: Ignore words that appear in less than this percentage of all documents. Range: real; 0.0-100.0

g) Prune above percent: Ignore words that appear in more than this percentage of all documents. Range: real; 0.0-100.0

h) Prune below absolute: Ignore words that appear in less than that many documents. Range: integer; 0-+?

i) Prune above absolute: Ignore words that appear in more than that many documents. Range: integer; 0-+?

j) Prune below rank: Specifies how many percent of the most infrequent words are ignored. Range: real; 0.0-100.0

k) Prune above rank: Specifies how many percent of the most frequent words are ignored. Range: real; 0.0-100.0

l) Data management: Determines, how the data is represented internally. Default: double_sparse_array

m) Select attributes and weights: If checked, you might select the used text attributes and their weights. Otherwise all text attributes are used. Range: Boolean; default: false

n) Specify weights: These parameters allow setting weights per attribute. Text from attributes with higher weight will be more important during analysis. Range: list

We checked "add meta information", prune method as absolute, prune below absolute as 4 and prune above absolute as 999 and data management as double_sparse_array.

3.2.2.1 Tokenize

This operator tokenizes a document.

This operator splits the text of a document into a sequence of tokens. There are several options to specify the splitting points. You may use all non-letter characters. This will result in tokens consisting of one single word. If you are going to build windows of tokens or something like that, you will probably split complete sentences, this is possible by setting the split mode to specify character and enter all splitting characters. The third option lets you define regular expressions and is the most flexible for very special cases. Each non-letter character is used as separator. As a result, each word in the text is represented by a single token [4].

Here we selected mode as non-letters.

3.2.2.2 Transform Cases

This operator transforms cases of characters in a document.

It transforms all characters in a document to either lower case or upper case, respectively. We selected lower case option. It takes input from 'tokenize' operator.

3.2.2.3 Filter Stop Words

This operator removes English stop words from a document.

It filters English stop words from a document by removing every token which equals a stop-word from the built-in stop word list. For this operator to work properly, every token should represent a single English word only. To obtain a document with each token representing a single word, you may tokenize a document by applying the text: tokenize operator beforehand. It takes input from 'transform cases' operator.

3.2.2.4 Generate N-Grams (Terms)

This operator creates term n-grams of tokens in a document.
A term n-gram is defined as a series of consecutive tokens of length n. The term n-grams generated by this operator consist of all series of consecutive tokens of length n.
It takes input from 'filter stop-words' operator. Max length parameter is considered as 3.

3.2.2.5 Filter Tokens by Length

This operator filters tokens based on their length.
It filters tokens based on their length (i.e. the number of characters they contain).
It takes input from 'Filter stop words' operator. Here min chars are considered as 2 and max chars as 999.

3.2.3 FP Growth

FP-growth works in a divide-and-conquer way [7]. The first scan of the database derives a list of frequent items in which items are ordered by frequency descending order. According to the frequency-descending list, the database is compressed into a frequent-pattern tree, or FP-tree, which retains the item-set association information. The FP-tree is mined by starting from each frequent length-1 pattern (as an initial suffix pattern), constructing its conditional pattern base (a "sub database", which consists of the set of prefix paths in the FP-tree co-occurring with the suffix pattern), then constructing its conditional FP-tree, and performing mining recursively on such a tree. The pattern growth is achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree.

The FP-growth algorithm transforms the problem of finding long frequent patterns to searching for shorter ones recursively and then concatenating the suffix. It uses the least frequent items as a suffix, offering good selectivity. Performance studies demonstrate that the method substantially reduces search time. extensions to the FP-growth approach, including depth-first generation of frequent item sets which explores a hyper-structure

mining of frequent patterns; building alternative trees; exploring top-down and bottom-up traversal of such trees in pattern-growth mining; and an array-based implementation of prefix-tree-structure for efficient pattern growth mining [7].

Advantages and disadvantages of FP-Growth

1) There is no candidate generation in FP-Growth.
2) FP tree eliminates repeated data scans.
3) Sometimes FP-Tree does not fit in memory.
4) Frequent item sets can be retrieved faster after FP Tree has been built.
5) It is expensive to build.

FP-Growth process can be described as follows.

This learner efficiently calculates all frequent item sets from the given data. It calculates all frequent items sets from a data set by building an FP-Tree data structure on the transaction data base. This is a very compressed copy of the data which in many cases fits into main memory even for large data bases. From this FP-Tree all frequent item sets are derived. A major advantage of FP-Growth compared to Apriori is that it uses only 2 data scans and is therefore often applicable even on large data sets.

Please note that the given data set is only allowed to contain binominal attributes, i.e. nominal attributes with only two different values. Simply use the provided preprocessing operators in order to transform your data set. The necessary operators are the discretization operators for changing the value types of numerical attributes to nominal and the operator nominal2binominal for transforming nominal attributes into binominal / binary ones. The frequent item sets are mined for the positive entries in your data base, i.e. for those nominal values which are defined as positive in your data base. This operator has two basic working modes: finding at least the specified number of item sets with highest support without taking the min_support into account (default) or finding all item sets with a support large than min_support. Here we considered min support as zero. File that we generated from document representation step is input file for this process.

3.2.4 Association Rules

This operator generates a set of association rules for a given set of frequent item sets. In RapidMiner, the process of frequent item set mining is divided into two parts: first, the generation of frequent item sets and second, the generation of association rules from these sets.

For the generation of frequent item sets, we used the operator FP Growth. The result will be a set of frequent item sets which could be used as input for this operator. Output is association rules.

All of these RapidMiner processes are repeated by gradually increasing number of keywords. Here are the steps we followed to do this –
Wordlist is generated by preprocessing of text. This wordlist has data of keywords as per total occurrence and occurrence in number of documents. This wordlist is manually examined to choose top 100 frequent keywords. We will omit general words from list like explains, describes etc.

We will start with 50 keywords. 50 keywords are queried in PubMed and abstracts are retrieved using same filters as described in section 3.1.1 and then fed to text preprocessing process as discussed in chapter 3.1. After preprocessing it is fed to RapidMiner and all the above operations are applied. After we get the results, increase the count of keywords by 5 Number of abstracts will increase as the number of keywords increases. This process is continued till we get the results without error. We go on increasing count till we get error on processing the data. Once we get the error, we have to decrease the count of keywords by 1 and check for errors. In summary, we have to find the maximum count of keywords for which we can run RapidMiner process without error. In our case, we got the results for maximum of 87 keywords.

3.2.5 Drawbacks of Exhaustive Frequent Pattern Mining by FP-Growth

FP-Growth algorithm generates all the frequent item sets from the data. But it is not very useful to generate frequent patterns exhaustively. Instead, it is always efficient to have only high quality and compact patterns in terms of time and space. As the number of keywords increases, frequent item set generation increase exponentially. As a result, it takes long time and after certain number of keywords we get error running the process. Hence interactive sampling algorithm by Mohammad Al Hasan, Snehasis Mukhopadhyay, Mansurul Buiyan [5] is a great solution which proposed novel approach to find interesting frequent patterns. Details of IPM are explained in next section.

3.3 Interactive Sampling Algorithm

In this section, we describe the interactive pattern mining approach in detail. This section is divided in following sections.

1) Introduction and Background
2) Text Preprocessing
3) Markov Chain, Metropolis-Hastings (MH) Algorithm
4) Interactive Sampling Algorithm
    4.1 Entity Selection
    4.2 Generate Neighbors
    4.3 User's Feedback
    4.4 Frequent Pattern Extraction
5) Advantages

3.3.1 Introduction and Background

In paper "Interactive pattern mining on hidden data: a sampling-based solution" by Mohammad Al Hasan, Snehasis Mukhopadhyay, Mansurul Buiyan [5], authors propose a solution to this problem that is based on Markov Chain Monte Carlo (MCMC) sampling of frequent patterns. The proposed algorithm doesn't return all the frequent patterns, it returns a small set of randomly selected patterns. It also allows interactive sampling, so

that the sampled patterns can fulfill the user's requirement effectively. Authors of paper consider the task of mining frequent patterns from a hidden dataset. Our thesis work is the extension of this paper research. Here we apply same concept to neuroscience domain. Main difference in thesis work and this research paper is that we are mining text data and it is related to neuroscience domain. In this thesis we generate all the input text data from scratch by querying PubMed database. Also, this paper randomly selects user feedback while this thesis asks for user's feedback as a console input. So program becomes more generic and there is no need to change program code if user needs to change his feedback values. This thesis work also compares the output of IPM with that of RapidMiner and tries to prove that IPM is more efficient in terms of space and time than exhaustive method.

Simply put, our system enables the user to provide feedbacks on the quality of the pattern that user receives; so in subsequent iterations, these feedbacks are used to bias the sampling towards the preferences of the user, so that the patterns obtained from subsequent samples are more interesting to the user.

3.3.2 Text Preprocessing

Input to interactive sampling algorithm requires certain text preprocessing operations. These operations are same as discussed in section 3.1 of text document preprocessing. Document extraction and document representation for IPM is already discussed in previous chapter 3.1.1 and 3.1.3 respectively. Input format as per Table 3.4 is used as input to IPM where numbers are mapped to keywords and total count of keywords is mentioned in last column.

3.3.3 Markov Chains, Metropolis-Hastings (MH) Algorithm

For mining frequent patterns we use Markov chain Monte Carlo (MCMC) based random walk on the frequent pattern space. Feedback of user on pattern is binary i.e. either pattern is interesting or pattern is not interesting. A Markov chain is the sequence of Markov process over the state space S. Markov chain is a mathematical system that

undergoes transitions from one state to another, between a finite or countable number of possible states [15].Transition probability matrix, T is used to guide the state-transition event. When the probability of being in any particular state is independent of the initial condition, Markov chain is said to reach stationary distribution $\prod$, Markov chain is reversible if there is probability distribution over states, $\prod$ such that

$\prod (i)\ T(i, j) = \prod (j)\ T(j, i), \forall i, j \in S.$

Here, $\prod$ is probability distribution,

T is transition probability matrix and i, j are states.

Positive and recurrent state has a period of 1 and it has finite mean recurrence time. Markov chain is ergodic if all states in an irreducible Markov chain are ergodic. A model is said to have ergodic property if there's a finite number N such that any state can be reached from any other state in exactly N steps [15]. If Markov chain has a stationary distribution it is ergodic. If vertices of a POG (partial order graph) are the state space (S) of a Markov chain and there is state transition only between adjacent nodes in POG, then Markov process acts as a random walk on the frequent pattern space [5].

Metropolis–Hastings (MH) algorithm is a Markov chain Monte Carlo (MCMC) method and it is used to draw a sequence of random samples from a probability distribution for which direct sampling is difficult. When the number of dimensions is more, MH algorithm is generally used.

MH algorithm can be used together with a random walk to perform Markov Chain Monte Carlo (MCMC) sampling. The Metropolis–Hastings algorithm generates a sequence of sample values in such a way that, as more and more sample values are produced, the distribution of values more closely approximates the desired distribution. It makes the sequence of samples into a Markov Chain as the distribution of next sample is dependent only on current sample value. In iteration next sample value is selected based on the current sample value. Then, if the sample is accepted its value is used in the next iteration and if not accepted current value is reused in the next iteration.

For this, the MH algorithm draws a sequence of samples from the target distribution as follows [5]:

i. Initialization - It picks an initial state x at random.

ii. Randomly pick a state y from current state x using a distribution q(x, y), referred as proposal distribution.

iii. Then, it calculates the acceptance probability.

It accepts the proposal move to y with probability ∏(x, y). This process is continued till Markov chain reaches to a stationary distribution. Here, the set of frequent patterns (F) is the state space of the random walk and the edges of POG (partial order graph) define the possible state transitions. The POG are generated locally as needed. Local neighborhood of pattern x consisting of all frequent super patterns and all frequent sub-patterns is constructed. As per figure 3.2 of acceptance probability, random walk goes to one of the neighbors. Each of the neighbors of the pattern x in POG is equally likely to be chosen. The acceptance probability for choosing the proposal move is as below:

$$\alpha(x,y) = \mathbf{min}\left(\frac{\gamma(y)\cdot(1/d_y)}{\gamma(x)\cdot(1/d_x)}, 1\right) = \mathbf{min}\left(\frac{\gamma(y)d_x}{\gamma(x)d_y}, 1\right)$$

Figure 3.2 Acceptance Probability to Choose Proposal Move

Here, y is the neighbor of the pattern x, γ is the target distribution. α (x,y) is the acceptance probability. dx and dy are the degrees of the pattern x and y in POG

3.3.4 Interactive Sampling Algorithm

In this thesis we use interactive sampling algorithm proposed by Mohammad Al Hasan, Snehasis Mukhopadhyay, Mansurul Buiyan [5]. Metropolis–Hastings (MH) algorithm is used as a sampling algorithm to generate frequent patterns. Below algorithm is from paper “Interactive pattern mining on hidden data: a sampling-based solution” by Mohammad Al Hasan, Snehasis Mukhopadhyay, Mansurul Buiyan [5].

We sample patterns using MH with a default sampling distribution, and update the default distribution using each of the user feedbacks so that the effective distribution converges towards with the updates.

```
Interactive_Sampling
(D, minsup, miniter, feedback_count, b):
1.  P = generate_any_frequent_pattern(D, minsup)
2.  dp = compute_degree(P)
3.  π(P) = compute_β-value(P)
4.  while (true)
5.     choose a neighbor, Q, uniformly from, all possible
       frequent super and sub patterns
6.     dQ = compute_degree(Q)
7.     π(Q) = compute_β-value(Q)
8.     acceptance_probability = min ((π(Q)dp)/(π(P)dQ), 1)
9.     if uniform(0, 1) ≤ acceptance_probability
10.       P = Q
11.       iter = iter + 1
12.    if(#ofNodeInCurrentPOG / #ofNodeInOldPOG >= 3)
13.       insert_random_edge()
14.    if iter % miniter == 0 feedback_count >=0
15.       get_and_process_feedback(P, b)
16.    else
17.       goto line 5

compute_degree(D, P, minsup):
1.  super_pat = all_frequent_super_pattern(P)
2.  sub_pat = all_sub_pattern(P)
3.  neighbor_list = super_pat ∪ sub_pat
4.  save the neighbor list in POG data structure
5.  return size(neighbor_list)

compute_β-value(P):
1.  for each single length p in pattern P
2.    value = value * w[p]
3.  return value

get_and_process_feedback(P, b):
1.  if feedback is positive
2.     for each single length p in pattern P
3.        w[p] = w[p] * b
4.  else
5.     for each single length p in pattern P
6.        w[p] = w[p] * 1/b
7.  feedback_count = feedback_count − 1
```

Figure 3.3 Interactive Sampling Algorithm

3.3.4.1 Entity Selection

Interactive sampling algorithm takes following inputs

1) Database D

2) minsup, a minimum support value, which defines whether a pattern is frequent or not

3) miniter, a minimum number of walks before the sampler returns a pattern to the user for feedback

4) feedback_count, total number of feedback and

5) b, a value for the base which is used to update weight of unit size pattern.

IPM system executes subroutine interactive_sampling for incoming user.

In Line 1, subroutine generate_any_frequent_pattern is executed with parameters database D and minimum support. It chooses any arbitrary frequent pattern P; a unit-length frequent pattern also works. Line 2 computes all frequent super and sub-patterns of P using subroutine compute_degree. The degree of P is the size of the union set of super and sub-patterns.

3.3.4.2 Generate Neighbors

After entity selection, beta values are calculated. In line 5, it chooses a pattern (Q) from P's neighbors uniformly. It then computes degree of Q and beta value of Q. Acceptance probability is calculated in Line 8. If user accepts the move, Q becomes the resident state; otherwise, the sampler chooses another neighbor and whole process is repeated. If the user aborts the sampling, the infinite while loop breaks. During the MCMC walk, if the condition in line 12, i.e., the ratio of the number of nodes in the current POG and the same in the previously marked POG is greater than 3, the sampler inserts a set of random edges to the current POG. Authors use a variable called random edge factor to control the number of inserted random edges. The value of random edge factor is not that important as long as a small fraction of edges in the state-transition graph are from the random graph. Here we keep this fraction equal to 0.20.

3.3.4.3 User's Feedback

The condition on line 14 checks whether the sampler should send the resident pattern (P) to the user for feedback. If condition succeeds, pattern is send to user for feedback. In this thesis we are giving option to input user feedback at the start of the program. Number of keywords of user's interest can be selected as console input.

3.3.4.4 Frequent Pattern Extraction

Depending on the feedback's status, the sampler updates the weight of each of the unit size sub-patterns of P, which forces a change in the target distribution.
In interactive pattern mining, the effective sampling distribution is t, which changes once a feedback on a pattern is received. We fix this interval (miniter) to be 25, i.e., the sampler returns a pattern to the user after it makes 25 walks. The base (b) has a large significance in regard to learning the user's desired distribution. We use b=1.75.

3.3.5 Advantages

1) Sampling-based pattern mining paradigm overcomes the information overload problem that is caused by the large number of frequent patterns in a traditional pattern mining task.
2) Pseudo-code for interactive sampling algorithm is generic and can easily be adapted for sampling any kind of patterns, e.g. item sets, trees, sequences or graphs.

CHAPTER 4. RESULTS

We tested RapidMiner FP-Growth algorithm and interactive sampling algorithm [5] to compare efficiency in terms of time and space. Objective of this thesis is to prove that interactive sampling algorithm is more efficient than RapidMiner FP-Growth algorithm which gives exhaustive frequent patterns. This chapter is divided in following sections

1. List of Frequent Keywords after Text Preprocessing
2. RapidMiner FP-Growth Results in Detail
    - 2.1 RapidMiner FP-Growth Input and Output
    - 2.2 RapidMiner FP-Growth Process Parameters
    - 2.3 Frequent Patterns by RapidMiner FP-Growth
    - 2.4 Association Rules using RapidMiner FP-Growth
    - 2.5 Constraints of RapidMiner
3. Interactive Pattern Mining Results in Detail
    - 3.1 IPM Input and Output
    - 3.2 Frequent Patterns by IPM
4. Summary
5. Visualization using Graphviz

4.1 List of Frequent Keywords after Text Preprocessing

Initially 25 keywords are selected for abstract extraction. Then as per section 3.1.2 frequent keywords are selected and these keywords are mapped to numbers for IPM system. The table on next page shows the frequent keywords and their mapping.

Below table shows how each keyword is mapped to the number. For example, number 1 is used for keyword ‘alcohol preferring rat’, number 46 is for ‘hydrolase’.

Table 4.1 Frequent Keywords and Mapping

Number	Keywords	Number	Keywords
1	Alcohol preferring rat	46	Hydroxylase
2	Addiction	47	Adenosine
3	Alcoholic hallucinosis	48	Hyperalgesia
4	Alcoholism	49	Acetylcholine
5	Anterior cingulate cortex	50	Mutations
6	Catechol-o-methyl-transferase	51	Astrocytes
7	Cognition	52	Olanzapine
8	Disulfiram	53	Clozapine
9	Dopamine	54	Cytokine
10	Gamma amino butyric acid	55	Fluoxetine
11	Gamma oscillation	56	Decarboxylase
12	Glutamate	57	Hypersensitivity
13	Hypofrontality	58	Hydroxytryptamine
14	Ketamine	59	Aripiprazole
15	Neonatal ventral hippocampal lesion	60	Phosphorylated
16	Nmda	61	Methyltransferase
17	Nmda hypofunction	62	Acetylaspartylglutamate
18	Nucleus accumbens	63	Cardiomyocytes
19	Orbitofrontal cortex	64	Prodynorphin
20	Prefrontal cortex	65	Depolarization

Table 4.1 Continued

Number	Keywords	Number	Keywords
21	Prelimbic cortex	66	Mitochondria
22	Schizophrenia	67	Prolactin
23	Sensitization	68	Deoxyglucose
24	Serotonin	69	Neuropeptide
25	Ventral tegmental area	70	Neurophysiological
26	Psychiatric	71	Endothelial
27	Neurons	72	Phosphatase
28	Therapeutic	73	Flupenthixol
29	Aspartate	74	Psychotropic
30	Nucleotide	75	Corticosterone
31	Mitochondrial	76	Epithelial
32	Dendritic	77	Delusions
33	Amphetamine	78	Paroxetine
34	Morphine	79	Methylphenidate
35	Hypothalamus	80	Nociception
36	Cerebral	81	Immunoglobulin
37	Neurotrophic	82	Endocannabinoid
38	Dehydrogenase	83	Antinociceptive
39	Glutathione	84	Catecholamines
40	Glucose	85	Tetrahydropyridine
41	Tryptophan	86	Monoamines
42	Synapses	87	Antigens
43	Norepinephrine	88	Peroxidase
44	Glycine	89	Benzodiazepine
45	Hyperactivity	90	Lipopolysaccharide

4.2 RapidMiner FP-Growth Results in Detail

4.2.1 RapidMiner FP-Growth Input and Output

Table 4.2 Input Parameters for FP-Growth

Input	
Number of keywords	87
Number of abstracts	1323636

Table 4.3 Output Parameters for FP-Growth

Output	
Number of frequent item sets	4913485 (maximum size - 14)

4.2.2 RapidMiner FP-Growth Process Parameters

Following figure shows steps for RapidMiner FP-Growth algorithm.

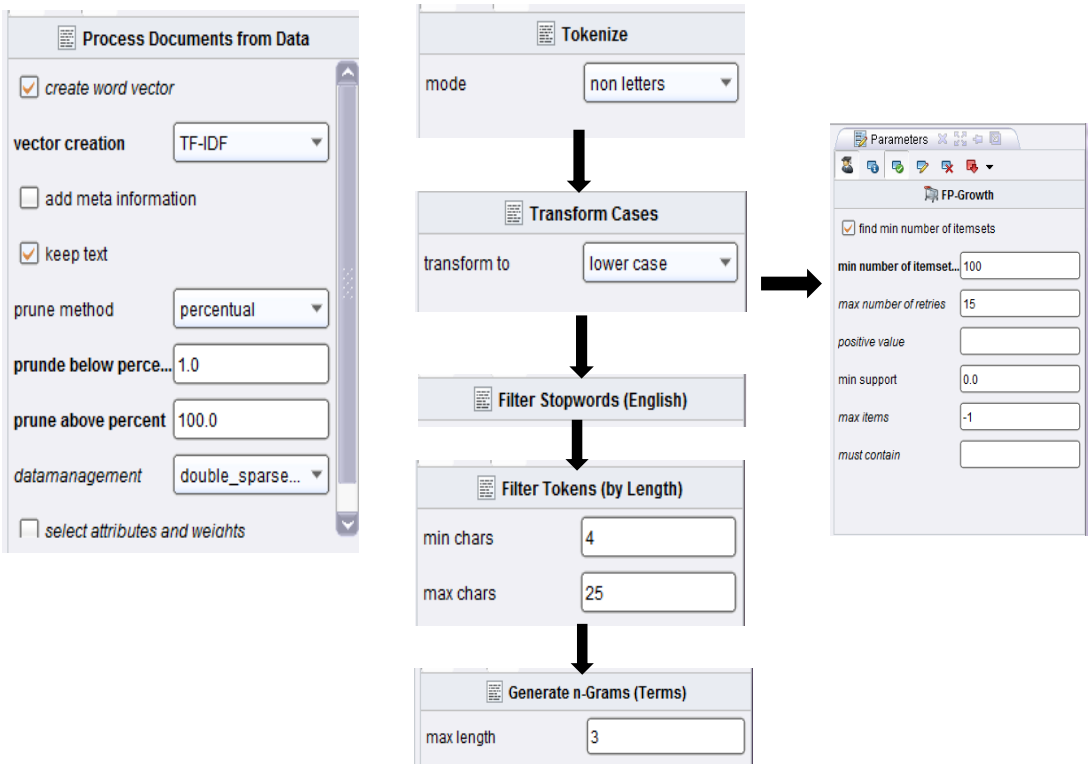

Figure 4.1 RapidMiner FP-Growth Steps

Total frequent patterns obtained are 4913485 and maximum size of pattern is 14. Because of large volume of output, only 4 frequent patterns are shown in below table.

Table 4.4 Frequent Patterns by RapidMiner FP-Growth

Frequent Patterns by RapidMiner FP-Growth					
Keywords		**1**	**2**	**3**	**4**
	Item 1	nucleotide	nucleotide	nucleotide	nucleotide
	Item 2	psychiatric	psychiatric	psychiatric	psychiatric
	Item 3	therapeutic	therapeutic	therapeutic	therapeutic
	Item 4	aspartate	aspartate	aspartate	aspartate
	Item 5	hyperalgesia	hyperalgesia	hyperalgesia	hyperalgesia
	Item 6	neurotrophic	neurotrophic	neurotrophic	neurotrophic
	Item 7	cerebral	cerebral	cerebral	cerebral
	Item 8	adenosine	adenosine	adenosine	adenosine
	Item 9	neurons	neurons	neurons	neurons
	Item 10	amphetamine	amphetamine	amphetamine	amphetamine
	Item 11	Dopamine	Dopamine	Dopamine	Dopamine
	Item 12	Addiction	hydroxylase	prolactin	Ketamine

4.2.4 Association Rules using RapidMiner FP-Growth

Following figure shows the association generated using RapidMiner. Figure shows only small portion of the output.

```
[Cognition, Schizophrenia, norepinephrine] --> [Dopamine] (confidence: 0.800)
[Cognition, addiction, cytokine] --> [Schizophrenia] (confidence: 0.800)
[Cognition, Glutamate, ventral_tegmental_area] --> [Dopamine] (confidence: 0.800)
[Cognition, Hypofrontality, prefrontal_cortex] --> [Dopamine] (confidence: 0.800)
[synapses, gamma_oscillation] --> [Schizophrenia] (confidence: 0.800)
[Schizophrenia, Glutamate, Serotonin ] --> [Nmda] (confidence: 0.800)
[Delusions, neurons, addiction] --> [Dopamine] (confidence: 0.800)
```

Figure 4.2 Association Rules using RapidMiner

4.2.5 Constraints of RapidMiner

RapidMiner's processing time increases as the number of keywords goes on increasing.

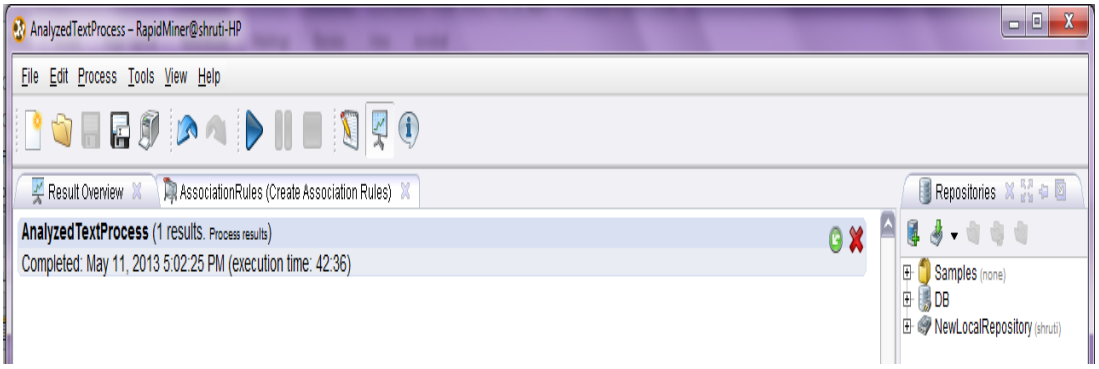

Figure 4.3 Time Required by RapidMiner FP-Growth

Thus, with 88 numbers of keywords, RapidMiner is unable to process the data and gives following error message. Hence we can't process more than 87 keywords with RapidMiner.

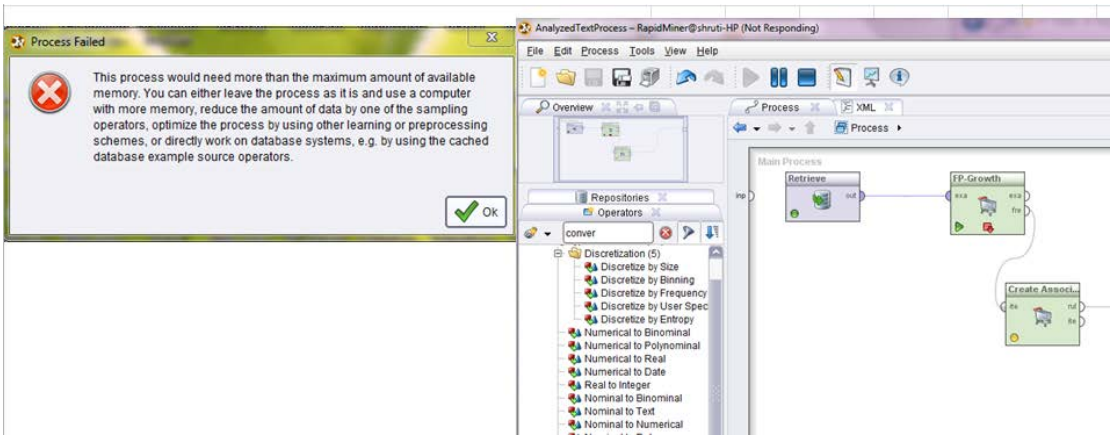

Figure 4.4 Error Message with 88 or More Number of Keywords

4.3 Interactive Pattern Mining Results in Detail

4.3.1 IPM Input and Output

Table 4.5 Input Parameters for IPM

Input	
Number of keywords	90
Number of abstracts	1323636
No of user-feedback keywords	5
Keywords of user-feedback	45,77,24,16,7
Minimum support	0
No of iterations	50
No of feedback iterations	50

Table 4.6 Output Parameters for IPM

Output	
Number of all frequent item sets	50
Number of unique frequent item sets	50

4.3.2 Frequent Patterns by IPM

Below figure shows small set of output (unique frequent patterns) where keywords are mapped to numbers. Last number of each row gives total of keywords in frequent patterns. Table 4.1 gives details about keywords and number mapping.

```
4_8_16_21_23_27_29_32_33_34_42_45_47_48_50_51_61_71_72_76_78_82_83_84_86_88_90_ 27
4_8_16_21_23_27_29_32_33_34_42_45_48_50_51_61_71_72_76_78_82_83_84_86_88_90_    26
4_8_16_21_23_27_29_32_33_34_42_45_48_50_51_61_71_72_78_82_83_84_86_88_90_       25
4_8_16_20_21_23_27_29_32_33_34_42_45_48_50_51_61_71_72_78_82_83_84_86_88_90_    26
4_8_16_20_21_23_27_29_32_33_34_42_45_48_50_51_61_71_72_78_82_83_84_85_86_88_90_ 27
4_8_16_20_21_23_26_27_29_32_33_34_42_45_48_50_51_61_71_72_78_82_83_85_86_88_90_ 27
4_8_16_20_21_23_26_27_29_32_33_34_42_45_50_51_61_71_72_78_82_83_85_86_88_90_    26
4_16_20_21_23_26_27_29_32_33_34_42_45_50_51_61_71_72_78_82_83_85_86_88_90_      25
4_16_20_21_23_26_27_29_32_33_34_42_45_50_51_61_69_71_72_78_82_83_85_86_88_90_   26
4_16_20_21_23_26_27_29_33_34_42_45_50_51_61_69_71_72_78_82_83_85_86_87_88_90_   26
4_16_20_21_23_26_27_29_33_34_42_45_50_51_61_69_71_72_78_82_85_86_87_88_90_      25
```

Figure 4.5 Unique Frequent Patterns by IPM

4.4 Summary

Interactive sampling algorithm is a probabilistic model to find user interesting patterns. This model learns to get the unknown user interesting patterns. Here user model is not predefined. IPM favors user's interest progressively by adjusting the sampling distribution using user's feedback. Text literature i.e. input dataset is very large and it is not useful and sometimes feasible to completely enumerate all the frequent patterns. So, the IPM uses the output space sampling (OSS) paradigm to obtain a frequent pattern in an interactive way.

Interactive sampling algorithm is based on Metropolis–Hastings algorithm. MH algorithm is a Markov chain Monte Carlo (MCMC) method for obtaining a sequence of random samples from a probability distribution for which direct sampling is difficult [30]. We use MH algorithm to mine patterns from the database, D. Each sampler runs an instance of MH sampling with the objective that MH's target sampling distribution (γ) matches with the desired sampling distribution of the user u. The Metropolis–Hastings algorithm generates a sequence of sample values in such a way that, as more and more sample values are produced, the distribution of values more closely approximates the desired distribution. These sample values are produced iteratively making sequence of

samples into a Markov chain. The distribution of the next sample is dependent only on the current sample value. The algorithm picks a candidate for the next sample value based on the current sample value. In our work, the sampler has no knowledge of the desired sampling distribution. So, we use default sampling distribution to sample patterns using MH and update the default distribution using each of the user feedbacks so that the effective distribution converges towards with the updates. To model γ (MH's target sampling distribution), we used a vector space model of unit-size patterns. We used similarity between interactive pattern mining and QEIR. QEIR is Query by Example in information retrieval. It means given a set of documents that a user has liked, retrieve other documents that he is likely to like [3] [5]. For IPM, the set of patterns for which the user provides a positive feedback plays a role that is similar to the role of the given documents in a QEIR task. QEIR uses the vector space model of words and phrases to construct user's preference profile [5]. In IPM, the effective sampling distribution is continuously updated by binary feedback from the user. The sampling distribution changes in response to the user's feedback performing random walk.

At initialization, all unit-size patterns have a weight of 1. The sampler starts sampling from the initial distribution by using the MCMC based random walk. The sampler picks a pattern P of length k for user's feedback. The pattern is discarded if it has already shared with user and the process continues until another pattern is found which is not yet sent to the user. User provides feedback as positive or negative. If the feedback is positive, the sampler increases the weight of each of k length-one sub-patterns of P exponentially. On the other hand, if the feedback is negative, the sampler decreases the weight of those length-one sub-patterns by dividing the current weight by b. This weight modification updates the existing sampling distribution. The sampling and the interaction steps are repeated intermittently as the function iteratively moves closer to the user's desired distribution. This weight update ensures that in all k, there exists a positive probability to visit any frequent pattern from any other frequent pattern [5].

In some techniques like constraint-based mining, a user can add additional constraints as an interactive input. The mining system considers these constraints to filter the output set to match it according to the user's requirements. Sometimes, setting constraint may work but sometimes it may yield sub-optimal results.

Here interactive sampling algorithm only relies on user's feedback to drive the pattern selection process, so it is not needed for the user to have access to the actual dataset which is considered for mining. Rather, the user exploits the interactive pattern mining system to obtain the desired patterns [5]. It overcomes the information overload problem that is caused by the generating exhaustive frequent patterns by traditional pattern mining. Table 4.7 shows the inputs and results of IPM and RapidMiner-FP Growth. As text data, a total of 1323636 documents were downloaded from PubMed containing 90 keywords listed in table 4.1. This download process takes around 2 hrs. Then MySQL database is created and desired input files are created using SQL scripts.

RapidMiner FP-Growth and interactive sampling algorithms are run and their results are compared. IPM gives 50 frequent patterns for 90 keywords, while RapidMiner gives 4913485 frequent patterns. Below table summarizes processing time and number of frequent patterns generated, for RapidMiner and interactive sampling algorithm.

Table 4.7 Summary for FP-Growth vs IPM

	RapidMiner	Interactive sampling algorithm
Number of terms	87	90
Number of documents	1323636	1323636
Time taken for querying and downloading abstracts from PubMed	2 hrs	2 hrs
Time taken for calculating frequent patterns	42.36 min	12 sec
No of frequent patterns generated	4913485	50

4.5 Visualization using Graphviz

Graphviz 2.28 is an open source graph visualization software. It is used to visualize the frequent patterns generated by IPM.

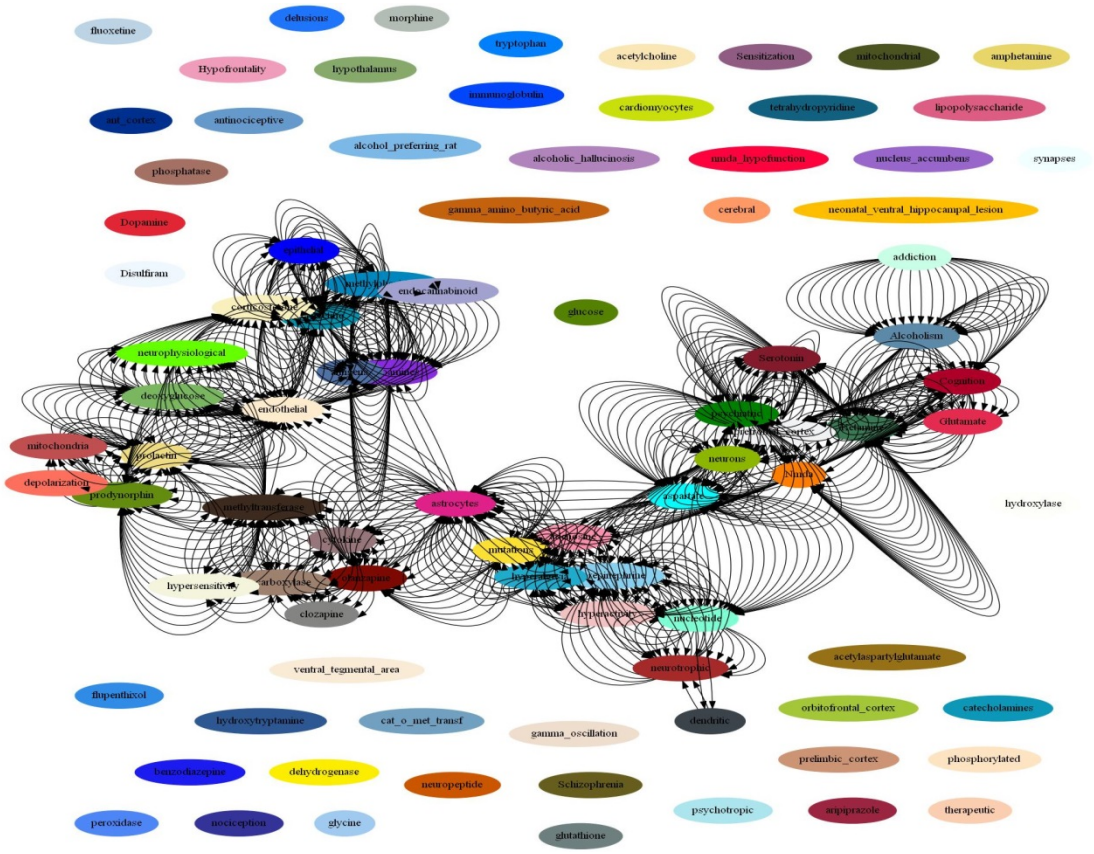

Figure 4.6 Visualization of Frequent Patterns by IPM using 50 Iterations

CHAPTER 5. CONCLUSION

In this thesis work we propose interactive approach to find frequent patterns among various biological keywords related to neuroscience appearing in the medical literature. This interactive sampling approach has been proven to be better than exhaustive frequent pattern mining FP-Growth approach using 90 keywords with 1323636 text abstracts. It has been found that interactive pattern mining (IPM) system generates compact but high quality patterns in very less time. As these patterns are based on user's feedback only useful patterns are retrieved instead of all patterns. This approach is personalized as user can change his feedback as per requirement. This results in both saving of time and space. Hence this IPM is more practical and useful approach. The patterns extracted could then be verified by experiment. Finding frequent patterns from medical literature can help researchers visualize previously unknown new frequent patterns among biological entities. In this system, there is no need to study each text documents manually to find associations between keywords, so minimum manual intervention. So this eliminates probability of human error while extracting knowledge from text.

Here we tested IPM approach on neuroscience data mainly related to disorders like schizophrenia and alcoholism. Abstracts are downloaded from PubMed. We compared the performance of our approach on 3 parameters: Time taken to generate frequent patterns, no of frequent patterns and maximum number of keywords processed.

The experimental results show that IPM takes 12 sec to give desired interesting output, while exhaustive method takes time in the range of 42 min. IPM is very efficient in terms of time than exhaustive. Number of frequent patterns by IPM are 50, while for exhaustive FP-Growth we get 4913485 frequent patterns. It is very difficult to study many frequent

patterns and derive a good research direction. Instead of that less but desired frequent patterns are more productive. Also exhaustive method's performance decreases in terms of time and space as number of keywords increases. In RapidMiner with 1323636 abstracts, we could only process till 87 keywords. We are getting error when number of keywords is greater than 87.While in this work IPM processed same number of abstracts with 90 keywords in seconds.

This shows that IPM performs better than exhaustive pattern mining approach. It is efficient in terms of time and space and usefulness. This method is particularly useful when the number of frequently co-occurring keywords is more.

Also, this approach is generic. The program code developed can be used to find frequent pattern for any keywords, not necessarily related to neuroscience or medical data. It just takes the input of keywords and returns the frequent patterns for those keywords. Thus we can change keywords of interest without changing the algorithm. This approach to find desired number of frequent patterns by changing value of iteration is one more positive aspect of this algorithm. This approach is scalable too. As text data is increasing rapidly nowadays, there is a need of more scalable approach to accommodate this growth. IPM approach is also cheaper than generating all possible combinations of these keywords and selecting the frequent ones out of them. Interactive sampling algorithm can be simultaneously run on x number of machines to perform x random walks. This parallelism helps to increase speed and efficiency.

Hence Interactive pattern mining algorithm [5] offers a novel mechanism to find interesting frequent patterns in literature. In future we hope to improve user feedback process by modifying the code so that user can give his feedback during execution of program. I believe this change will add more flexibility to the program. One more improvement we can do is in mode of user's feedback. Instead of current binary feedback, if user can give feedback in range of values or scale, it will improve performance and generate more accurate results.

The frequent patterns can be used to gain knowledge about the keywords and their context. This thesis work of frequent pattern mining of neuroscience data will be very useful for medical researchers. It may play a big role in research for the treatments of disorders, schizophrenia and alcoholism.

REFERENCES

REFERENCES

[1] "Text Mining," [Online]. Available: http://en.wikipedia.org/wiki/Text_mining. [Accessed 5 August 2012].

[2] "Natural Language Preprocessing," [Online]. Available: http://en.wikipedia.org/wiki/Natural_language_processing. [Accessed 18 February 2013].

[3] "TF-IDF," [Online]. Available: http://en.wikipedia.org/wiki/Tf%E2%80%93idf. [Accessed 25 April 2013].

[4] "RapidMiner," [Online]. Available: http://en.wikipedia.org/wiki/RapidMiner. [Accessed 29 November 2012].

[5] M. Hasan, S. Mukhopadhyay and M. Buiyan, "Interactive Pattern Mining on Hidden Data : A Sampling-based Solution," in CIKM, Hawaii,US, 2012.

[6] C. Lapish, S. Mukhopadhyay and N. Tirupattur, "Text Mining for Neuroscience: A Co-morbidity Case Study," in Knowledge-Based Systems in Biomedicine and Computational Life Science, Springer Berlin Heidelberg.

[7] J. Han, H. Cheng and D. Xin, "Frequent Pattern Mining: Current Status and Future," Data Mining and Knowledge Discovery, vol. 15, no. 1, 2007.

[8] Neuroscience of Psychoactive Substance Use and Dependence, World Health Organization, 2004.

[9] "Addiction," [Online]. Available: http://en.wikipedia.org/wiki/Addiction. [Accessed 5 December 2012].

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

[11] A. Zanasi, Text Mining and its Applications to Intelligence, CRM and Knowledge Management, WIT Press, 2005.

[12] M. Palakal, S. Mukhopadhyay and M. Stephens, "Identification of Biological Relationships from Text Documents," in Medical Informatics - Knowledge Management and Data Mining in Biomedicine, 2005.

[13] "Rapid-I," [Online]. Available: http://rapid-i.com/content/view/181/190/. [Accessed 1 December 2012].

[14] F. Verhein, "Frequent Pattern Growth (FP-Growth) Algorithm," [Online]. Available: http://csc.lsu.edu/~jianhua/FPGrowth.pdf. [Accessed 05 March 2013].

[15] "Markov Chain," [Online]. Available: http://en.wikipedia.org/wiki/Markov_chain. [Accessed 12 December 2012].

[16] R. Drake and K. Mueser, "Co-Occurring Alcohol Use Disorder and Schizophrenia," Schizophr Bulletine, p. 616–617, 2006.

[17] "Neuroscience," [Online]. Available: https://en.wikipedia.org/wiki/Neuroscience. [Accessed 10 January 2013].

[18] J. Zhang, J. Mostafa and H. Tripathy, "Information Retrieval by Semantic Analysis and Visualization of the Concept Space of d-lib magazine," D-Lib Magazine, vol. 8, no. ISSN 1082-9873, 2002.

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

[20] S. Mukhopadhyay and H. Vaka, "Knowledge Extraction and Extrapolation Using Ancient and Modern Biomedical Literature," IEEE BioCom Workshop, 2009.

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

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

[23] M. Palakal, M. Stephens, S. Mukhopadhay and R. Raje, "Identification of Biological Relationships from Text Documents using Efficient Computational Methods," Journal of Bioinformatics and Computational Biology, Vols. 1,2, 2003.

[24] M. Hasan and M. Zaki, "Output Space Sampling for Graph Patterns," Proceedings of the VLDB Endowment, vol. 2, no. 1, 2009.

[25] M. Berry and M. Castellanos, Survey of Text Mining II - Clustering, Classification, and Retrieval, Springer-Verlog London Limited, 2008.

[26] M. Berry and J. Kogan, Text Mining Applications and Theory, John Wiley & Sons, Ltd, 2010.

[27] R. Feldman and J. Sanger, The Text Mining Handbook - Advanced Approaches in Analyzing Unstructured Data, Cambridge University Press, 2007.

[28] "PubMed," [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed. [Accessed 9 January 2013].

[29] B. Goethals and J. Bussche, "Interactive Constrained Association Rule Mining," [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.6.2288. [Accessed 12 December 2012].

[30] "Metropolis Hastings Algorithm," [Online]. Available: http://en.wikipedia.org/wiki/Metropolis%E2%80%93Hastings_algorithm. [Accessed 29 November 2012].

[31] T. Chia, K. Sim, H. Li and H. Ng, "A Lattice-based Approach to Query- by-Example Spoken Document Retrieval," in 31st ACM SIGIR conference on Research and development in information retrieval, 2008.

[32] P. Jörgensen, "Continuous analysis of internet text by artificial neural network," [Online]. Available: http://2011.ispace.ci.fsu.edu/~pjorgensen/pjorgensendiss.pdf. [Accessed 10 May 2012].