UNLV Theses, Dissertations, Professional Papers, and Capstones

5-2009

Text Categorization Based on Apriori Algorithm's Frequent Itemsets

Prathima Madadi
University of Nevada, Las Vegas

Follow this and additional works at: https://digitalscholarship.unlv.edu/thesesdissertations

Part of the Computer Engineering Commons, and the Systems and Communications Commons

Repository Citation

TEXT CATEGORIZATION BASED ON APRIORI ALGORITHM'S

FREQUENT ITEMSETS

by

Prathima Madadi

Bachelor of Technology in Computer Science and Engineering
Jawaharlal Nehru Technological University, India
May 2006

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science Degree in Computer Science
School of Computer Science
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
May 2009

UMI Number: 1472429

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI®

UNLV
UNIVERSITY OF NEVADA LAS VEGAS

Thesis Approval
The Graduate College
University of Nevada, Las Vegas

JANUARY 9TH , 20 09

The Thesis prepared by

PRATHIMA MADADI

Entitled

TEXT CATEGORIZATION BASED ON APRIORI ALGORITHM'S FREQUENT ITEMSETS

is approved in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Examination Committee Chair

Dean of the Graduate College

Examination Committee Member

Examination Committee Member

Graduate College Faculty Representative

ABSTRACT

Text Categorization Based on Apriori Algorithm's Frequent Itemsets

by

Prathima Madadi

Dr. Kazem Taghva, Examination Committee Chair
Professor of Computer Science
University of Nevada, Las Vegas

Automatic Text categorization is the task of assigning an electronic document to one or more categories, based on its contents. There are many known techniques to efficiently solve categorization problems. Typically these techniques fall into two distinct methodologies which are either logic based or probabilistic. In recent years, many researchers have tried approaches which are a hybrid of these two methodologies.

In this thesis, we deal with document categorization using Apriori Algorithm. The Apriori algorithm was initially developed for data mining and basket analysis applications in the relational databases. Although the technique is logic based, it also relies on the statistical characteristics of the data. As a part of this work, we will implement all the tools which are necessary to carry out automatic categorization using Apriori algorithm. We will also report on the categorization effectiveness by applying this technique to standard collections.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

CHAPTER 1

INTRODUCTION

Text categorization is the process of automatically categorizing documents to one or more predefined categories. It has witnessed a booming interest in the last two decades. Although the concept of text categorization came into existence in early 60's, it was widely known only in early 90's. Over the years it became one of the most challenging and widely researched areas, because of the increased availability of documents in digital form and the subsequent need to organize them [1].

A closely related area of categorization is Information Retrieval which deals with discovery of relevant information for user's queries. Major goal of information retrieval is to satisfy user's information needs. In other words it deals with the representation, storage, organization of and access to information items [3]. In recent years information retrieval and machine learning researchers are adopting text categorization as one of their applications of choice.

Text categorization is a supervised machine learning technique. It has become one of the key techniques for handling and organizing data because arranging documents manually is not only difficult but also time consuming and expensive. Moreover this interest is also due to the fact that text categorization

techniques have reached accuracy levels that can outperform even the trained professionals. This is achieved with high level of efficiency on standard software/hardware resources [2].

Text categorization has many diverse applications. Some of them are indexing of scientific articles according to predefined thesauri of technical terms, automated population of hierarchical catalogues of web resources, spam filtering i.e. detecting spam email messages by looking at the message header and content, identification of document genre, automated essay grading, categorization of news paper ads, grouping of conference papers into sessions, categorizing news stories as finance, weather, entertainment and sports [2].

Categorization is also used in the field of medical sciences to predict tumor cells as malignant or benign based on the results of MRI scan, in finance sector to determine credit card transactions as legitimate or fraudulent, and also in the study of astronomical objects to categorize galaxies as spiral or elliptical based on their shape as shown in Figure 1.1 [10, 16].
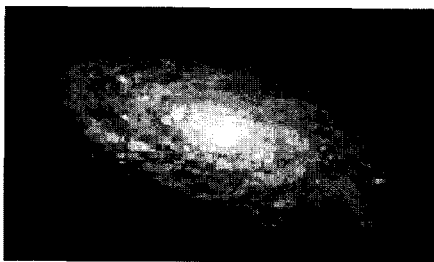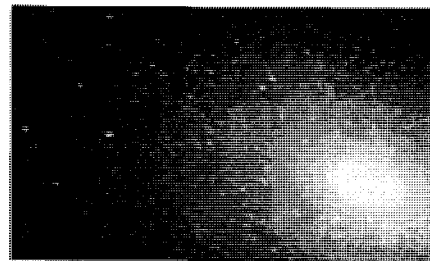
(a) A spiral galaxy   (b) An elliptical galaxy

Figure 1.1. Categorization of galaxies.

This thesis deals with automatic categorization based on apriori Algorithm. Apriori algorithm developed by Agarwal and Srikant [11] is a well known algorithm in data mining with applications in market basket transactions analysis. Instead of market basket transaction, this thesis concentrates on a basket of significant terms retrieved from a collection of text documents which are consequently used in training of the categorizer. Once training phase is completed, this apriori based categorization engine is used to predict category labels of documents it has not seen during training phase. We further evaluate the effectiveness of this technique by calculating its precision and recall on a test collection.

1.1 Thesis Structure

This Thesis is organized into six chapters including the introduction chapter. Chapter 2 presents the background of categorization giving details of naive Bayes categorization based on Bayes theorem. In chapter 3 a clear explanation of frequent itemsets generation is illustrated. Chapter 4 presents implementation details and experimental results of this thesis. Chapter 5 evaluates the results presented in chapter 4. Chapter 6 concludes this thesis by giving a brief description about future proceedings.

CHAPTER 2

BACKGROUND

Data stored in computer files and databases is increasing at a phenomenal rate. Users working on these data are more interested to extract useful information from them, rather than using the entire data. A marketing manager working for a grocery store is not satisfied with just a list of all items sold, but wants a clear picture of what customers have purchased in the past as well as predictions of their future purchases. Data mining thus evolved to meet these increasing information demands [4].

Data Mining is defined as the process of extracting previously unknown, useful information from databases. In recent years data mining not only attracted business organizations, but also has been widely used in the information technology industry. Data mining is playing an important role in real world applications due to the availability of large amounts of data, and need to turn that data in to useful information. There are many well known data mining tasks, categorization is one among them on which this thesis concentrates. Categorization is a supervised machine learning technique [4, 5].

Machine Learning is defined as "the ability of a machine to improve its performance based on previous results" [6]. In other words it is a system capable of learning from experience and analytical observation, which results in

continuous self improvement there by offering increased efficiency and effectiveness [7]. In general there are four different types of machine learning techniques. They are:

1. Supervised learning.

2. Unsupervised learning.

3. Semi-supervised learning and

4. Reinforcement learning [8].

This thesis deals with text categorization which is a supervised learning technique.

Supervised learning: Supervised learning is a machine learning technique that learns from training data set. A training data set consists of input objects, and categories to which they belong. Assigning categories to input objects is carried out manually by an expert. Given an unknown object, supervised learning technique must be able to predict an appropriate category based on prior training.

2.1 Categorization

Categorization is one of the most popular and familiar data mining techniques.

Definition: Given a database $D = \{ t_1, t_2, \ldots\ldots\ldots\ldots t_n \}$ of objects and a set of categories, $C = \{ C_1, C_2 \ldots\ldots\ldots\ldots, C_n \}$, the problem of categorization is to define a mapping $f: D{\rightarrow}C$ where each item t_i is assigned to one category. A category C_j, contains only those objects mapped to it; that is,

$C_j = \{ t_i \mid f(t_i) = C_j , 1 \leq i \leq n \text{ and } t_i \in D \}$ [4].

Categorization can also be defined as "the task of learning a target function f that maps each object set x to one of the predefined class labels y" as shown in Figure 2.1 [10, 16].
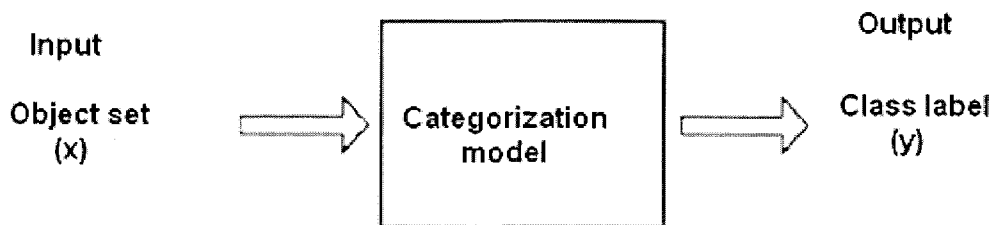
Figure 2.1.Categorization mapping input object set x to class label y.

The target function is also known as a categorization model. A categorization model helps in distinguishing between objects of different classes. Input data for a categorization task is a collection of records. Each record is characterized by a tuple (x, y) where x is the object set and y is designated as a class label known as a category. Table 2.1 shows a vertebrate training data set for classifying vertebrates into one of the following categories like mammal, bird, fish, reptile, or amphibian. Here x an object set includes properties of a vertebrate such as its name, body temperature, type of reproduction, ability to fly and ability to live in water. Object set as shown in Table 2.1 are mostly discrete but in general they can contain continuous features, where as category label must always be a discrete object.

Table 2.1. The vertebrate training data set.

Name	Body Temperature	Gives Birth	Aquatic Creature	Aerial Creature	Category Label
Human	Warm-blooded	yes	No	No	Mammal
Turtle	Cold-blooded	No	Yes	No	Reptile
Frog	Cold-blooded	No	Yes	No	Amphibian
Bat	Warm-blooded	Yes	No	Yes	Mammal
Pigeon	Warm-blooded	No	No	Yes	Bird

Categorization model built from the above data set is used to predict categories of unknown records. When an object set with a new record is given to the categorization model, it can be treated as a black box, which automatically assigns a category label to that record. In detail, categorization technique should be able to predict the correct category label based on previous training. To illustrate this, consider a new vertebrate creature 'whale' as a new record shown in Table 2.2. Based on previous training, categorization model should be able to predict the appropriate category to which creature 'whale' belongs to? [10].

Table 2.2. The vertebrate test set.

Name	Body Temperature	Gives Birth	Aquatic Creature	Aerial Creature	Class Label
Whale	Warm-blooded	Yes	Yes	No	?

2.2 General Approach to Solve a Categorization Problem

Categorization technique builds categorization models from an input data set. For this process it should first choose a learning algorithm. The learning algorithm must build a model that best fits the relationship between object set and categories of the input data. This model must also predict the categories for new records which are previously unknown. Figure 2.2 shows a general approach for solving categorization problems [16]. Initially for any categorization problem a collection of data set is given. This data set is further divided in to a training data set and a test data set.

Training set: A training set is a collection of records whose categories are known. This set is used in building categorization model as discussed above, which is then applied to the test set.

Test set: A test set is a collection of records whose categories are known. Categorization model must predict categories for these known records. Test set determines accuracy of categorization model based on the count of test records correctly and incorrectly predicted [10].

8

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Test Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?
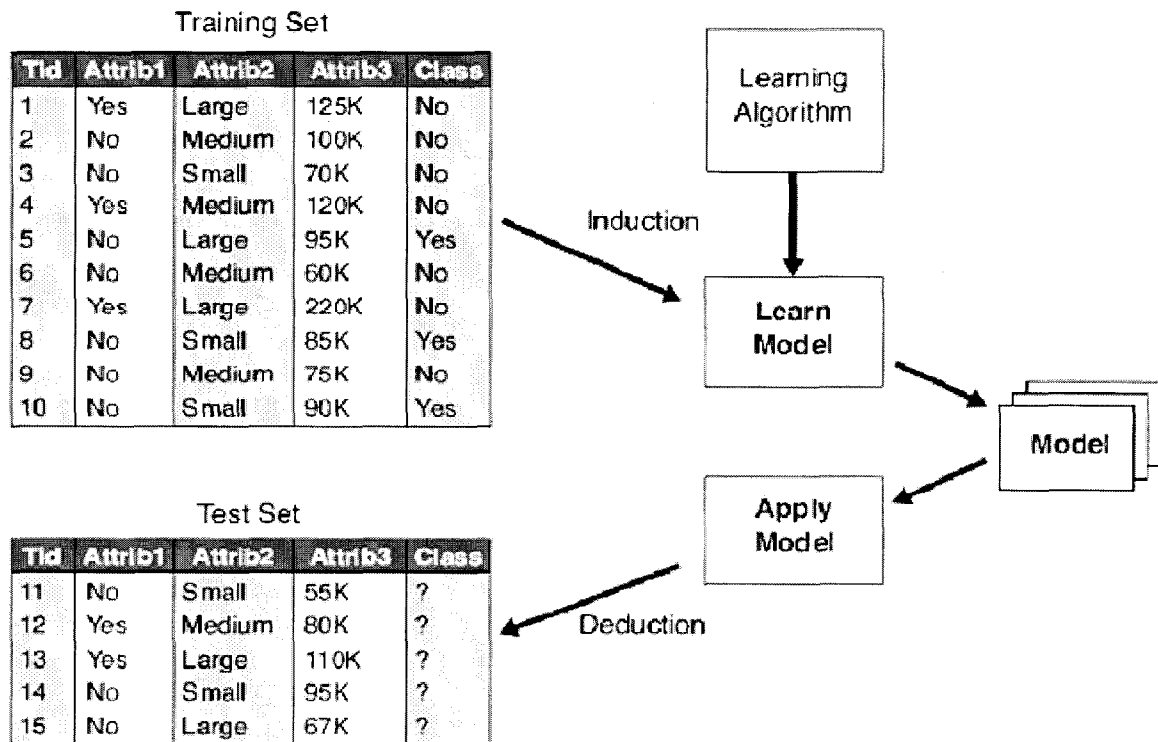
Figure 2.2. General approach for building a categorization model.

There are many standard categorization methods in use. They are:

1. Decision tree categorization.

2. Rule based categorization.

3. Neural networks.

4. Support vector machine.

5. K nearest neighbor.

6. Bayesian categorization.

From the categorization methods available, Bayesian is one of the most well known

categorization technique [9].

2.3 Bayesian Categorization

Bayesian Categorization is used to predict class membership probabilities i.e. probability of a given sample belongs to a particular category [9]. It is based on Bayes theorem. The term "Bayes" refers to the reverend English mathematician Thomas Bayes. "Bayes Theorem is a simple mathematical formula used for calculating conditional probabilities" [12].

2.3.1 Bayes Theorem

Let X be a data sample whose category is unknown. Let H be some hypothesis say data sample X belongs to a specified category C. For categorization problems one need to determine P(H | X) the probability that the hypothesis H holds given the observed data sample X.

Bayes theorem is given by:

$$P(H \mid X) = \frac{P(X \mid H)\, P(H)}{P(X)}$$

Where P (H | X) is the posterior probability of H conditioned on X. For example, consider a data sample consisting of fruits described by their color and shape. Suppose that X is red and round, and that H is the hypothesis that X is an apple. Then P (H | X) implies that X is an apple given that, it is observed to be red and round. P (H) is the prior probability of H i.e. regardless of what the data sample looks; it is the probability that the given sample is apple. Posterior probability is based on information such as background knowledge rather than the prior probability which is independent of data sample X.

In the same way, P (X | H) is the posterior probability of X conditioned on H i.e. probability that X is red and round, and it is true that X is an apple. P (X) is the prior probability of X. It is the probability that a data sample from the set of fruits is red and round [9].

Given a large data sample, it would be difficult to calculate above probabilities. To overcome this difficulty, conditional independency was introduced.

2.4 Naive Bayes Categorization

Naive Bayes categorization is a simple probabilistic Bayesian categorization [13]. It assumes that the effect of an attribute value on a given category is independent of the values of other attributes. This assumption is called conditional independence which was introduced to simplify complex computations involved, hence the name "naive". It exhibits high accuracy and speed when applied to large databases, and its performance is comparable with decision trees and neural networks.

Step wise representation of naive Bayes categorization:

1. Initially each data sample is represented as a vector, $X = (x_1, x_2,........., x_n)$ which are measurements made on the sample from n attributes, respectively, $A_1, A_2,.....,A_n$.

2. Suppose that there are m categories, $C_1, C_2,..........,C_m$. If an unknown data sample X is given, then the categorization model will predict the correct category for X based on highest posterior probability, conditioned on X. Naive

11

Bayes categorization will assign unknown sample X to the class C_i if and only if

$$P(C_i \mid X) > P(C_j \mid X) \text{ for } 1 \le j \le m, j \ne i.$$

Where

$$P(C_i \mid X) = \frac{P(X \mid C_i)\, P(C_i)}{P(X)} \quad (\text{By Bayes Theorem})$$

3. As $P(X)$ is constant for all classes, only $P(X \mid C_i)\, P(C_i)$ need to be calculated. If the prior probabilities of categories are not known, then it can be assumed that all are equally likely i.e. $P(C_1) = P(C_2) = \ldots\ldots\ldots = P(C_m)$. Prior probabilities of categories can be calculated by $P(C_i) = s_i / s$, where s_i is the number of training samples of class C_i and s is the total number of training samples.

4. It is extremely expensive to compute $P(X \mid C_i)$ for data sets with many attributes. In order to reduce this computation naive Bayes categorization assumes conditional independence. By this assumption values of the attributes are conditionally independent of one another given the category of the sample. There are no dependence relationships among the attributes. Thus,

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i).$$

5. If an unknown sample X is given then the naive Bayes categorization computes the value of $P(X \mid C_i)\, P(C_i)$ for each category. Unknown sample X is then assigned to the category C_i if and only if

$$P(C_i \mid X) P(C_i) > P(C_j \mid X) P(C_j) \text{ for } 1 \le j \le m, j \ne i.$$

In other words categorization model maps sample X with the category C_i having

maximum P (C_i | X)P (C_i) value [9].

2.4.1 Predicting a Category Using Naive Bayes Categorization

Consider a training data set that describes weather conditions for playing some unspecified game as shown in Table 2.3. Data sample is represented by a set of attributes such as outlook, temperature, humidity, windy and categories by attribute play. Play is represented as either "Yes" or "No". Consider C_1 has optimistic category for play and C_2 as pessimistic category for play. Each data sample is represented as a vector. There are nine vectors which belong to category 'Yes', and five vectors that belong to category "No" from a total of fourteen vectors.

Suppose an unknown sample X = (sunny, cool, high, true) is given. The model computes to which category X belongs by calculating P (X | play = "Yes") P(play="Yes") and P (X | play = "No")P (play = "No"). Sample X is mapped to category having maximum posterior probability. Initially prior probability for each category can be computed based on the training sample. A naive Bayes categorization model can now be built from the training data set as shown below.

Table 2.3. Training dataset that describes weather conditions.

Outlook	Temp	Humidity	Windy	Play
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rainy	mild	high	false	Yes
rainy	cool	normal	false	Yes
rainy	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rainy	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rainy	mild	high	true	No

P (play = "Yes") = 9/14 = 0.642   (See step 3 of naive Bayes categorization).

P (play = "No") = 5/14 = 0.357

Conditional probabilities for sample X are calculated as follows:

P (sunny | Yes), P (cool | Yes), P (high | Yes), P (true | Yes),

P (sunny | No), P (cool | No), P (high | No) and P (true | No).

P (sunny | Yes) = 2/9

P (cool | Yes) = 3/9

P (high | Yes) = 3/9

P (true | Yes) = 3/9

P (sunny | No) = 3/5

P (cool | No) = 1/5

P (high | No) = 4/5

P (true | No) = 3/5

Using the above probabilities, we obtain

P (X | play = "Yes") = 2/9 * 3/9 * 3/9 * 3/9

$$= 0.0082$$

P (X | play = "No") = 3/5 * 1/5 * 4/5 * 3/5

$$= 0.0576$$

P (play="Yes" | X) = P (X | play= "Yes") P (play= "Yes")

$$= 0.0082 * 0.642$$

$$= 0.0053$$

P (play="No" | X) = P (X | play = "No") P (play = "No")

$$= 0.0576 * 0.357$$

$$= 0.0206.$$

Categorization model will assign sample X to category play= 'No' because probability of P (play="No" | X) is greater than probability of P (play="Yes" | X). Therefore, the naive Bayes categorization maps sample X to category "No" [14].

CHAPTER 3

APRIORI ALGORITHM'S FREQUENT ITEMSETS GENERATION

Apriori invented by Rakesh Agarwal and Ramakrishnan Srikant [11] in 1994 is

a well known algorithm in data mining. It was originally applied to market basket

transactions. Instead of market basket transactions, this thesis work is based on

a basket of significant terms obtained from a collection of electronic documents.

This chapter illustrates frequent itemsets generation of the Apriori algorithm

by taking a general transaction database example as shown in Table 3.1. Each

row in the table represents a transaction, which contains unique transaction

identification number (TID) along with items bought by the customer represented

as {A, B, C, D, E}.

Table 3.1. The transaction database.

TID	Items
1	{A, B, C}
2	{A, B, C, D, E}
3	{A, C, D}
4	{A, C, D, E}
5	{A, B, C, D}

The transaction database can be represented in binary form of 0's and 1's as shown in Table 3.2. Each row corresponds to a transaction and each column corresponds to an item. If an item exists in a transaction then it is represented as '1' otherwise '0' [15].

Table 3.2. A binary representation of transaction database.

TID	A	B	C	D	E
t1	1	1	1	0	0
t2	1	1	1	1	1
t3	1	0	1	1	0
t4	1	0	1	1	1
t5	1	1	1	1	0

3.1 Definitions

Let $T = \{t_1, t_2, \ldots, t_N\}$ be the set of all transactions and $I = \{i_1, i_2 \ldots i_d\}$ be the set of all items in a transaction database. Each transaction t_i consists of items which are subsets of I.

Itemset: It is defined as a collection of zero or more items in a transaction. If an itemset has no items in it then it is termed as a null itemset, and if it contains k items then it is referred as a k-itemset.

Support count: Support count is defined as the number of transactions that contain a particular itemset. It is the most important property of an itemset.

17

Mathematically given by:

$$\sigma(X) = | \{t_i \mid X \subseteq t_i , t_i \in T \} |$$

Where |.| indicates the number of elements in the set.

To illustrate this consider a 2-itemset say {A, B} from Table 3.1. Support

count is 3 because there are only three transactions that contain itemset

{A, B}.

Support: It is defined as how often an itemset is applicable to a given dataset.

Formally given by:

$$Support, s = \frac{supportcount}{N}$$

Where

N = Number of transactions in the database [10].

Consider the example shown above for calculating support. Support count is 3

and total number of transactions is 5 as shown in Table 3.1. So,

$$s = \frac{3}{5} = 0.6 \ .$$

3.2 Frequent Itemsets Generation

Itemsets that satisfy minsup are considered as frequent itemsets i.e. support

of an itemset must satisfy the user specified support threshold.In general, a

dataset containing k items can generate up to $2^k - 1$ frequent itemsets excluding

the null itemset. Figure 3.1 shows a lattice structure that lists all possible itemsets

for I = { a, b, c, d, e} including the null itemset [16].

18
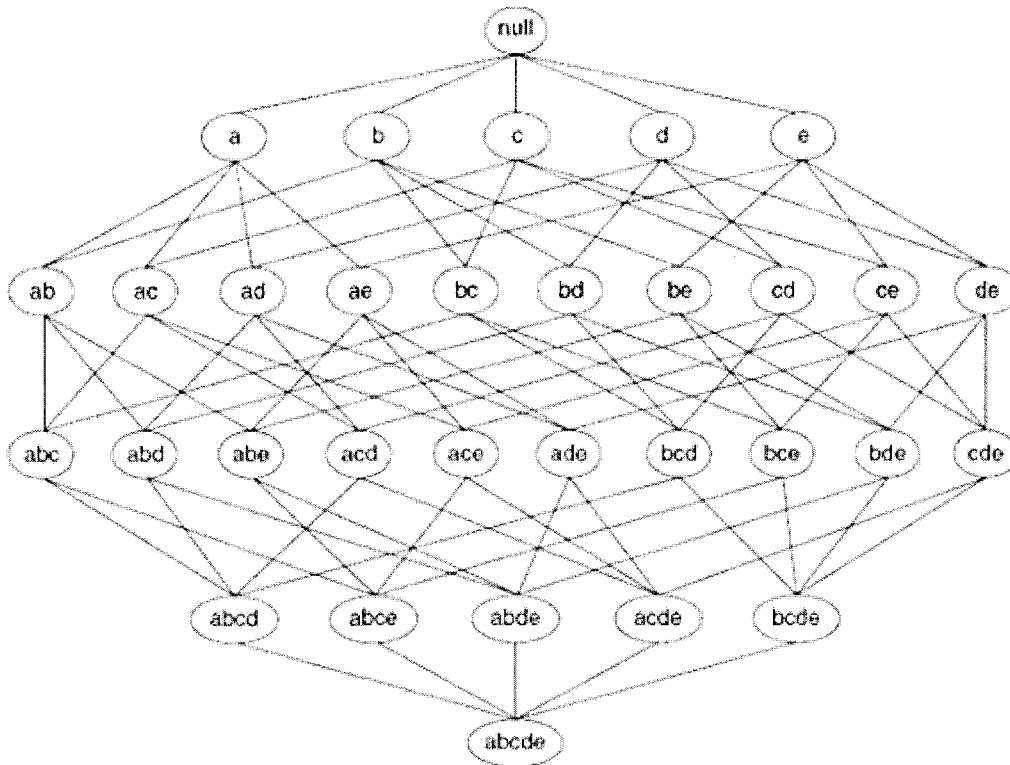
Figure 3.1. An itemset lattice.

3.2.1 The Apriori Principle

Theorem: If an itemset is frequent, then all of its subsets must also be frequent [10].

This can be illustrated by considering an itemset lattice as shown in Figure3.2 [16]. Suppose itemset {c, d, e} is a frequent itemset, then all of its subsets {c}, {d}, {e}, {c, d}, {c, e} and {d, e} must also be frequent because any transaction that contain {c, d, e} must also contain its subsets.

19
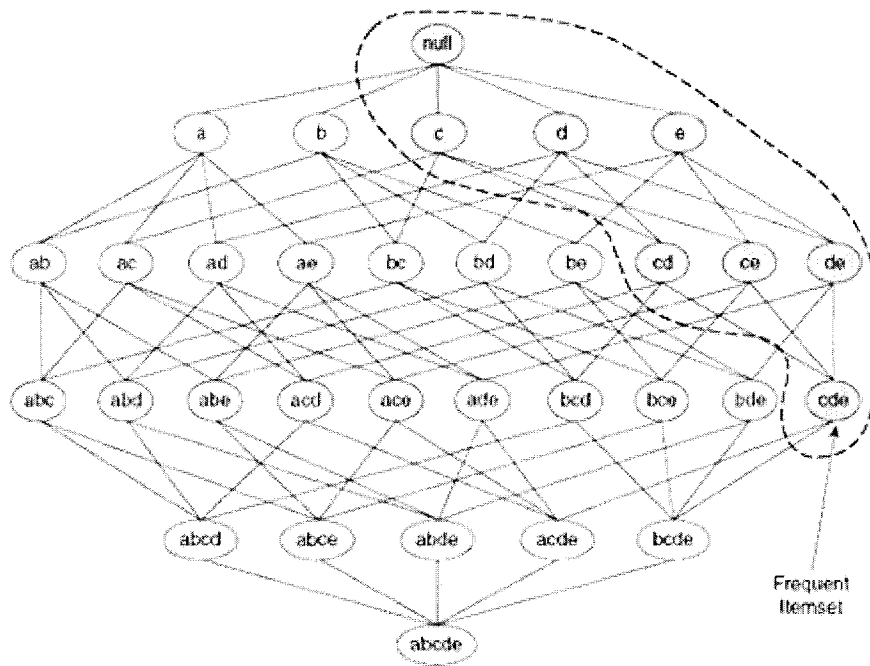
Figure 3.2. An illustration of Apriori principle – If an itemset is frequent then all its subsets are also frequent.

Conversely, if an itemset say {a, b} is found to be infrequent then all of its supersets {a, b, c}, {a, b, d}, {a, b, e}, {a, b, c, d}, {a, b, c, e}, {a, b, d, e} and {a, b, c, d, e} must also be infrequent. So, {a, b} along with all its supersets can be pruned as shown in Figure 3.3 [16]. This method of trimming search space based on support value is called support-based pruning. The key property behind this is "that support for an itemset never exceeds the support for its subsets also known as anti-monotone property". Apriori was the first mining algorithm that uses support-based pruning to reduce the exponential growth of candidate itemsets. A candidate itemset is defined as a potential frequent itemset [10].
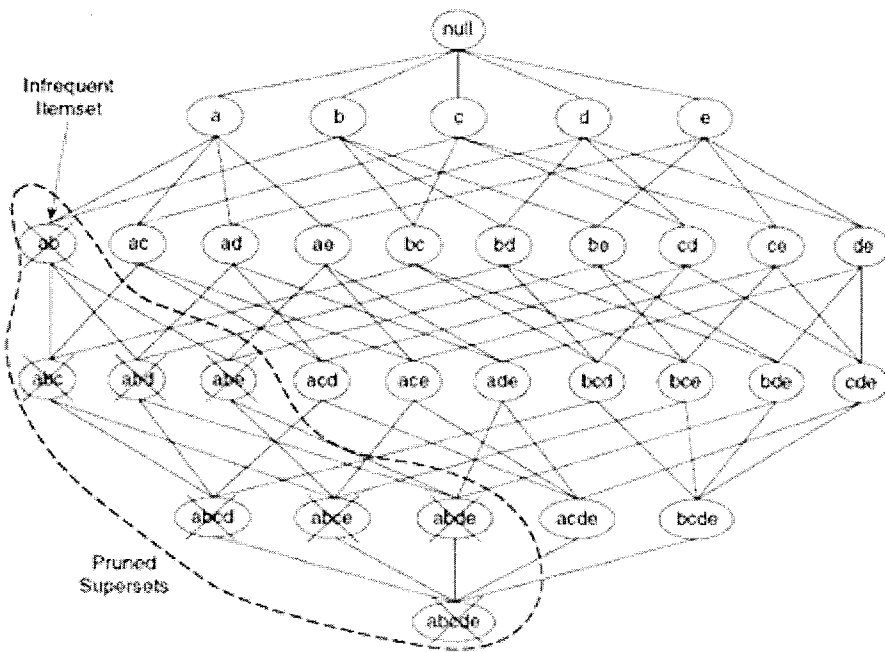
Figure 3.3. An illustration of support-based pruning- If an itemset is infrequent then all its supersets are also infrequent.

3.2.1.1 Apriori Algorithm Pseudo Code

Pseudo code for generating frequent itemsets in Apriori algorithm is presented below [15]:

Pass 1

1. Generate the candidate itemsets in C_1.

2. Save the frequent itemsets in L_1.

Pass k

1. Generate the candidate itemsets in C_k from the frequent itemsets in L_{k-1}.

   (i). Join $L_{k-1}p$ with $L_{k-1}q$, as follows:

   insert into C_k

select $p.item_1$, $q.item_1$, . . . , $p.item_{k-1}$, $q.item_{k-1}$

from $L_{k-1}p$, $L_{k-1}q$

where $p.item_1 = q.item_1$, . . . , $p.item_{k-2} = q.item_{k-2}$, $p.item_{k-1} < q.item_{k-1}$.

(ii). Generate all $(k\text{-}1)$-subsets from the candidate itemsets in C_k.

(iii). Prune all candidate itemsets from C_k where some $(k\text{-}1)$-subset of the

candidate itemset is not in the frequent itemset L_{k-1}.

2. Scan the transaction database to determine the support count for each

candidate itemset in C_k.

3. Save the frequent itemsets in L_k.

The algorithm is divided into two passes. In pass 1 all itemsets are

considered as candidate 1-itemsets. After finding their support counts only those

itemsets that satisfy minimum support count are saved as frequent 1-itemsets.

In pass k, the algorithm iteratively generates new candidate k-itemsets using

the frequent (k-1)-itemsets found in the previous iteration using $F_{k-1} * F_{k-1}$ method

which is explained below.

$F_{k-1} * F_{k-1}$ Method: Candidate k- itemsets are generated by merging a pair of

frequent (k-1)-itemsets only if their first (k-2) items are identical.

For example let A = {a_1, a_2,....., a_{k-1}} and B = { b_1, b_2,....., b_{k-1} } be a pair of

frequent (k-1)-itemsets. A and B can be merged only if they satisfy the following

condition:

$a_i = b_i$ (for i = 1, 2,..........., k-2) and $a_{k-1} < b_{k-1}$.

Once this step is completed then candidate pruning is performed to eliminate

some of the candidate k-itemsets. Consider a candidate k-itemset, say

22

$X = \{i_1, i_2...., i_k\}$. The algorithm must check whether all of its proper subsets are frequent i.e. $X - \{i_j\}$ are frequent. If one of them is infrequent then X is immediately discarded. This will help in reducing the number of candidate itemsets considered in the next step of the algorithm-support counting.

Support counting is the process of determining the frequency of occurrence for all itemsets once they survive the candidate pruning step. The algorithm needs to make an additional pass over the database to calculate their support counts. Candidates that satisfy minimum support count are saved as frequent itemsets. The algorithm halts when there are no more candidate itemsets to generate [10]. The algorithm is clearly illustrated step-wise with an example using transaction database as shown in Table 3.1. Let us assume support threshold to be 40%, which is equivalent to a minimum support count of 2.

Pass 1:

Initially the algorithm assumes each item in the transaction database to be a candidate 1-itemset as shown in Table 3.3.

Table 3.3. Candidate 1- itemsets.

Itemset	Support count
A	?
B	?
C	?
D	?
E	?

After candidate 1-itemsets are generated their support counts are calculated. Only those itemsets are saved that satisfy the minimum support count known as frequent itemsets. Here nothing is been discarded as all itemsets satisfy the minimum support count which is 2 as shown in Table 3.4.

Table 3.4. Frequent 1-itemsets.

Itemset	Support count
A	5
B	3
C	5
D	4
E	2

Pass 2:

Candidate 2-itemsets are generated based on frequent 1-itemsets as shown in Table 3.5.

Table 3.5 Candidate 2-itemsets.

Itemset	Count
{A, B}	?
{A, C}	?
{A, D}	?
{A, E}	?
{B, C}	?
{B, D}	?
{B, E}	?
{C, D}	?
{C, E}	?
{D, E}	?

Nothing is pruned since all subsets of the candidate 2-itemsets are frequent. Support count is calculated for each candidate 2-itemset as shown in Table 3.6.

Table 3.6. Support count for candidate 2-itemsets.

Itemset	Support count
{A, B}	3
{A, C}	5
{A, D}	4
{A, E}	2
{B, C}	3
{B, D}	2
{B, E}	1
{C, D}	4
{C, E}	2
{D, E}	2

As {B, E} is infrequent all of its supersets are also infrequent from the property of support-based pruning. So, it is discarded and remaining itemsets are saved as frequent 2-itemsets.

Pass 3:

To generate candidate 3-itemsets look at the first (k-2) frequent 2-itemsets. If they satisfy the condition discussed in F_{k-1} * F_{k-1} method, then merge them as shown in Table 3.7.

Table 3.7. Candidate 3- itemsets.

	Itemset	Support count
Join AB with AC	{A, B, C}	?
Join AB with AD	{A, B, D}	?
Join AB with AE	{A, B, E}	?
Join AC with AD	{A, C, D}	?
Join AC with AE	{A, C, E}	?
Join AD with AE	{A, D, E}	?
Join BC with BD	{ B, C, D}	?
Join CD with CE	{C, D, E}	?

During the candidate pruning step itemset {A, B, E} is eliminated because its subset {B, E} is infrequent. After calculating their support count frequent 3-itemsets are saved as shown in Table 3.8.

Table 3.8. Frequent 3-itemsets.

Itemset	Support count
{A, B, C}	3
{A, B, D}	2
{A, C, D}	4
{A, C, E}	2
{A, D, E}	2
{ B, C, D}	2
{C, D, E}	2

Pass 4:

Candidate 4-itemsets are generated by checking whether the first (k-2) items of frequent 3-itemsets are equal. Only two itemsets satisfy this property as shown in Table 3.9.

Table 3.9. Candidate 4-itemsets.

	Itemset	Support count
Join ABC with ABD	{A, B, C, D}	?
Join ACD with ACE	{A, C, D, E}	?

In the candidate pruning step nothing is eliminated because for itemset {A, B, C, D} all its subsets {A}, {B}, {C}, {D}, {A, B}, {A, C}, {A, D}, {B. C},{B, D} and {C, D} are frequent. Same holds for itemset {A, C, D, E}.So, the corresponding frequent 4-itemset saved is shown in Table 3.10.

Table 3.10. Frequent 4-itemsets.

Itemset	Support count
{A, B, C, D}	2
{A, C, D, E}	2

Pass 5:

No candidate 5-itemsets are generated because there are no frequent 4-itemsets beginning with the same three items. Hence the algorithm halts [15]. Categorization of electronic documents using frequent itemsets is explained clearly in the next chapter.

CHAPTER 4

TEXT CATEGORIZATION USING FREQUENT ITEMSETS

The goal of this research is to categorize electronic documents to one or more categories, based on frequent itemsets and to determine efficiency of the categorization model built. This thesis concentrates on the study and implementation of the model presented in [17].

To perform text categorization, a collection of electronic documents are obtained from the "Reuters-21578, Distribution 1.0 test collection". There are 21578 newswire stories from Reuters, classified into several sets of categories, by personnel from Reuters Ltd. and Carnegie Group, Inc in 1987. It was further formatted by David D. Lewis and Peter Shoemaker in 1991. Reuters-21578 collection divided in to five sets with a total of 674 categories as shown in Table 4.1 [18]

.

Table 4.1. Reuters-21578 categories.

Field	Categories
Topics	135
Organizations	56
Exchanges	39
Places	176
People	269

As many other researchers before, this thesis work also concentrates on Topics set. Out of 135 categories available in Topics set, only five categories are chosen to run this experiment.

They are:

1. Acquisition,

2. Grain,

3. Interest Rate,

4. Jobs and

5. Trade.

There are a total of 504 documents from the collection that are mapped to these five categories. These documents are further divided into two sets. They are

1. Training set with 304 documents and

2. Test set with 200 documents.

Training set collection is shown in Table 4.2 and Test set collection is shown in section 4.2.2.

Table 4.2. Training set collection.

Category	Total number of documents
Acquisition	70
Grain	60
Interest Rate	70
Jobs	34
Trade	70

All documents are represented in "Standard Generalized Markup Language" format. Documents vary in length such as one to more than 50 lines and in number of categories assigned i.e. none to more than one category. Figure 4.1 is a screenshot of document in SGML format from Reuters 21578 collection.

```
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="18418" NEWID="2000">
<DATE> 5-MAR-1987 09:05:42.45</DATE>
<TOPICS><D>jobs</D></TOPICS>
<PLACES></PLACES>
<PEOPLE></PEOPLE>
<ORGS></ORGS>
<EXCHANGES></EXCHANGES>
<COMPANIES></COMPANIES>
<UNKNOWN>
A RM
f09771reute
b f BC-U.S.-FIRST-TIME-JOBLE   03-05 0080</UNKNOWN>
<TEXT>
<TITLE>U.S. FIRST TIME JOBLESS CLAIMS FALL IN WEEK</TITLE>
<DATELINE>   WASHINGTON, March 5 - </DATELINE><BODY>
New applications for unemployment
insurance benefits fell to a seasonally adjusted 332,900 in the
week ended Feb 21 from 368,400 in the prior week, the Labor
Department said.The number of people actually receiving benefits under
regular state programs totaled 3,014,400 in the week ended Feb
14, the latest period for which that figure was available.
That was up from 2,997,800 the previous week.
reuterend
</BODY></TEXT>
</REUTERS>
```

Figure 4.1. A screenshot of category Jobs

Each document starts with a Reuters tag and ends with a Reuters tag as shown in Figure 1.1.The topics tag indicates to which category the document belongs manually categorized by experts. Text of the story is enclosed in the body tag and every story ends with a reuterend statement. A clear description of tags is given in [18].

31

4.1 Documents Processing

Initially, all training documents are parsed to remove markup tags and special formatting using a parser [20]. The Implementation of this thesis is coded in Java. The output of parser is just the content inside the body tag. Once documents are parsed they should be tokenized.

Tokenization is the process of breaking parsed text into pieces, called tokens [21]. During this phase text is lowercased and punctuations are removed. For example consider the sentence "Although there was inflation, at least the economy worked," from a document that belong to category Trade tokenized as shown in Table 4.3.

Table 4.3. List of tokens.

although
there
was
inflation
at
least
the
economy
worked

Next step after tokenization is removing stop words. Common words such as 'are', 'the', 'with', 'from' etc. that occur in almost all documents, does not help in deciding whether a document belongs to a category or not. Such words are

referred as stop words. So, these words can be removed by forming a list of stop words. This thesis works on a total of 416 stop words.

Before removing stop words there are a total of 52034 terms in all 304 training documents, but after removing stop words there are reduced to 27910 terms including duplicates. Thus, 24124 words are removed which appeared to be of little value saving both space and time. Once stop words are removed, next step performed is stemming.

Stemming refers to the process of reducing terms to their stems or root variant. For example "computer", "computing", "compute" is reduced to "comput" and "engineering", "engineered", "engineer" is reduced to "engine". The main advantage of using stemming is to reduce computing time and space as different forms of words are stemmed to a single word. The most popular stemmer in English is the Martin Porter's stemming algorithm shown to be empirically effective in many cases [19]. It is implemented in various programming languages which are available for free. This thesis works on stemming algorithm programmed by porter in java [22]. After stemming all terms, next step is to build an inverted index.

Inverted Index is an index data structure storing a mapping from content, such as terms to its locations in a set of documents [23]. There are two types of inverted index, where in this thesis concentrates on record level inverted index. A record level inverted index contains a list of references to documents for each term. Consider a simple example with three documents say D_1, D_2 and D_3.

D_1: "it is an apple"

D$_2$: "apple is in the basket"

D$_3$: "it is a banana"

A record level inverted index file built is shown in Table 4.4. For the term "apple" document Id is represented as 1 and 2 because it occurs in both documents D$_1$ and D$_2$.

Table 4.4. A record level inverted index file.

Term	Document Id
apple	{1, 2}
basket	{2}
banana	{3}

In this thesis inverted index is built along with document frequencies to figure out significant terms in the collection. Document frequency is defined as the number of documents that contain a particular term. Consider the above example for which document frequencies are shown in Table 4.5. Document frequency for the term "apple" is 2 because it occurs in two documents D$_1$ and D$_2$. A sample screenshot of training terms along with their document frequencies is shown in Figure 4.2.

Table 4.5. Terms with their document frequencies.

Term	Document Frequency
apple	2
basket	1
banana	1

```
accept17
access13
accompani4
accord9
account15
achiev6
acquir27
acquisit17
acreag5
act6
action14
activ13
ad46
add9
addit10
adjust25
administr19
admit4
adopt5
advanc9
advantag5
advisori5
affair8
affect10
africa4
afternoon5
ag9
agenc16
aggreg4
ago13
agre33
agreement48
agricultur43
ahead8
aid18
aim14
air4
alloc4
allow14
altern4
```

Figure 4.2. A sample screenshot of terms along with their document frequencies.

Document frequency of term "administr" is 19 because it occurs in 19 documents out of 304. Total number of terms obtained after building inverted index is 3608 excluding all duplicates.

A major difficulty of text categorization problems is the high dimensionality of feature space i.e. total number of terms considered. Even for a moderate-sized text collection there are hundreds of thousands of unique terms [24]. So, our concentration is to reduce the number of terms in the collection which is referred as dimensionality reduction. There are many known methods to perform dimensionality reduction. This thesis works on term selection based on document frequency thresholding. Document frequency thresholding is the simplest dimensionality reduction technique used for reducing vocabulary in the collection. This is carried out based on a predefined threshold value such that only those terms are removed from the collection which are less than the given threshold value.

This thesis concentrates only on those terms whose document frequency is greater than three and less than 90 and excludes the remaining terms. Suppose, if document frequency is less than three then those terms are considered as rare terms as they appear in fewer documents. Basically, rare terms are considered to be non-informative for category prediction in global performance and hence can be removed [22]. If terms have document frequency greater than 90, it means that these terms occur in almost half of the document. So, by these terms one cannot distinguish between two documents and hence can be removed. After

removing all these terms that does not satisfy predefined threshold value there are only 1025 terms left out of 3608 terms.

Once significant terms are obtained, the next step is to generate frequent itemsets. Frequent itemsets are generated using Apriori algorithm inorder to categorize documents into categories. A clear explanation of Apriori algorithm along with its pseudo code is presented in chapter 3. Instead of using a basket of items this work is carried out based on a basket of significant terms obtained from training documents. Transaction database is a collection of 304 documents. Items are denoted as significant terms and basket of terms are referred as an itemset.

Two files are given as input to the Apriori program. One is a config file and other is a transa file. Config file keeps track of the number of significant terms obtained, number of training documents considered and user specified minimum support threshold, where as transa file contains documents and terms in m x n matrix form, where rows represent document numbers and columns represent significant terms. If a term occurs in a document then it is represented as '1' otherwise '0' as explained in chapter 3. Our goal is to discover frequent itemsets inorder to categorize documents into categories. Minimum support threshold is defined as 5% i.e. if an itemset occurs in atleast 15 documents then only it is considered as a frequent itemset. Candidate 1-itemsets are generated directly from the document frequency table, candidate 2-itemsets are generated based on frequent 1-itemsets as shown earlier in chapter 3. This process continues until there are no more candidate itemsets to be considered. Here terms are

represented as numbers from one to 1025 in transa matrix and later converted to words. A sample output of Apriori program with frequent itemsets represented in numbers as well as words is shown below:

Frequent 1-itemsets:

2, 6, 8, 9,14, 17................

accept, account, acquir, acquisit, ad, adjust.......................

Frequent 2-itemsets:

{8, 168}, {8, 817}, {14, 76}, {14, 93}.............

{acquir, company}, {acquir, share}, {ad, bank}, {ad,billion}.................

Frequent 3-itemsets:

{76, 134, 718}, {76, 222, 292}, {76, 222, 412}.................

{bank, central, rate}, {bank, cut, effect}, {bank, cut, half}.................

4.2 Itemsets Categorization Method

In general categorization problem can be divided into two phases as explained in chapter 2. They are:

1. Training phase and

2. Test phase or Categorization phase.

4.2.1 Training Phase

In training phase, a set of documents along with their categories are defined by an expert. Then, a categorization model is built as explained in chapter 2. In this thesis categorization model is a java program which is trained using frequent itemsets.

Frequent itemsets are represented using 'π' based on their cardinalities such as 1-itemsets are represented as π_1, π_2,......... π_{N1}, 2-itemsets as π_{N1+1}, π_{N1+2}, ...π_{N1+N2}, 3-itemsets as $\pi_{N1+N2+1}$, $\pi_{N1+N2+2}$ ....$\pi_{N1+N2+N3}$. Etc. For each frequent itemset π, find all documents that contain this particular itemset. Let's designated these set of documents as Dπ. For example itemset π_1 corresponds to $D\pi_1$, π_2 corresponds to $D\pi_2$ etc. A sample screenshot of frequent 1, 2, 3-itemsets along with their documents are shown in Figure 4.3.

For each category C_i, there are a certain number of documents that fall in to this category. Such as documents that fall into category Trade are represented as DC_1, category Grain are represented as DC_2, category Interest are represented as DC_3, category Acquisition are represented as DC_4 and category Jobs are represented as DC_5 as shown below.

Trade = DC_1= {D1, D2, D3, D4......................, D70}

Grain = DC_2 = {D71, D72, D73, D74......................., D130}

Interest = DC_3 = {D131, D132, D133, D134,..................., D200}

Acquisition =DC_4 = {D201, D202, D203, D204,..................., D270}

Jobs = DC_5 = {D271, D272, D273, D274,....................,., D304}

Our goal is to determine which itemsets fall into which categories. Itemset π_j is mapped with category C_i based on the maximum value of $W_{\pi j}$.

. The weight $W_{\pi j}$ is calcuated based on the formula:

$$W_{\pi j} = D\pi_j \cap DC_i / DC_i \quad \text{where i = 1, 2, 3, 4, 5 categories.}$$

```
2
D9
D45
D76
D103
D114
D116
D128
D144
D158
D161
D163
D185
D189
D191
D224
D233
D254
end
6
D1
D8
D18
D30
D31
D65
D106
D119
D141
D143
D166
D178
D185
D191
D192
end
```

Figure 4.3. (i) Frequent 1-itemsets along with their documents.

```
8 168
D19
D201
D202
D203
D210
D217
D221
D226
D227
D234
D241
D242
D243
D251
D255
D260
D262
D263
end
8 817
D201
D202
D210
D217
D222
D226
D227
D231
D234
D235
D251
D253
D260
D262
D263
end
```

(ii) Frequent 2-itemsets along with their documents.

```
167 287 317
D12
D18
D20
D22
D24
D31
D42
D73
D83
D116
D117
D128
D129
D138
D298
D303
end
169 350 480
D53
D59
D69
D73
D96
D169
D185
D194
D275
D280
D281
D284
D285
D288
D299
D303
end
```

(iii) Frequent 3-itemsets along with their documents.

Denominator DC_i is used for normalizing with the number of documents associated with category C_i. It takes into account whether an itemset occurs in other categories as well. Significance of terms occurring frequently in documents other than DC_i is thus suppressed [17]. For example consider frequent 1-itemset as shown below:

Frequent 1-itemset = π_1 = {2} = {D9, D45, D76, D103, D114, D116, D128, D144, D158, D161, D163, D185, D189, D191, D224, D233, D254}.

To determine to which category this itemset can be mapped is by finding common documents between π_1 and DC_1, π_1 and DC_2, π_1 and DC_3, π_1 and DC_4 and π_1 and DC_5. π_1 is mapped only with that category which has maximum $W_{\pi j}$ value.

$W_{\pi 1} = D\pi_1 \cap DC_1 / DC_1 = 2/70 = 0.028$.

$W_{\pi 1} = D\pi_1 \cap DC_2 / DC_2 = 4/60 = 0.083$.

$W_{\pi 1} = D\pi_1 \cap DC_3 / DC_3 = 7/70 = 0.1$.

$W_{\pi 1} = D\pi_1 \cap DC_4 / DC_4 = 3/70 = 0.042$

$W_{\pi 1} = D\pi_1 \cap DC_5 / DC_5 = 0/34 = 0.0$.   .

Hence, itemset π_1 is associated with category Interest because it has the highest weight when compared to associating this itemset with other categories. In the same way weights for itemsets2 and itemsets3 are constructed. All categories are mapped with their representative itemsets based on $W_{\pi j}$ values. Category Trade along with its representative itemsets is shown below in Figure 4.4.

ad
administr
agreement
american
annual
associ
bill
billion
call
chairman
chief
competit
ad billion
ad cut
ad export
ad foreign
ad industri
ad intern
ad major
ad month
ad state
agreement countri
billion countri
billion deficit
billion econom
billion end
billion export
countri state unit
export foreign state
export state unit
foreign good state
foreign japan surplu
foreign state unit
billion countri foreign
billion export import
billion foreign state
billion foreign surplu
billion import surplu
billion state unit

Figure 4.4. A screenshot of itemsets belonging to category Trade.

In this way every category is mapped with their representative itemsets. During testing phase the model based on these representative itemsets must classify the new unseen documents to correct categories. This is known as a supervised learning technique as the model is trained based on predefined documents and their categories.

4.2.2 Test Phase:

Whenever a new document is given the categorization model must predict correct category label based on previous training.

As there are frequent 1-itemsets, 2-itemsets etc. a weight factor, wf is defined to distinguish between singles, pairs, triplets of an itemset i.e. 1-itemsets are defined by wf_1, pairs by wf_2, triplets by wf_3 etc. Higher the cardinality higher the weight factor.

A model associates new document to the correct category based on the below formula:

$$W_{cj} = \sum_{i=1}^{C_j} Wf_{\pi i}$$

Where $(\pi_i \in C_j)$ $\wedge$ $(\pi_i \subseteq D)$, for all j = 1, 2, 3, 4, 5 categories.

D is the set of significant terms obtained from the new test document.

$Wf_{\pi j}$ is the weight factor of frequent itemsets.

Categorization weight is determined by the sum of weight factors for all itemsets of a given category [17]. Test document is associated with only that category which has maximum weight factor. A collection of test documents is given in Table 4.6.

45

Table 4.6. Test set collection

Category	Total Documents
Trade	58
Grain	4
Interest	56
Acquisition	70
Jobs	12

Automatic categorization of a test document is shown by taking an example

shown in Figure 4.5.

```
<REUTERS TOPICS="YES" LEWISSPLIT="TEST" CGISPLIT="TRAINING-SET" OLDID="4154" NEWID="15171">
<DATE> 8-APR-1987 13:19:26.77</DATE>
<TOPICS><D>trade</D></TOPICS>
<PLACES><D>usa</D><D>japan</D></PLACES>
<PEOPLE></PEOPLE>
<ORGS></ORGS>
<EXCHANGES></EXCHANGES>
<COMPANIES></COMPANIES>
<UNKNOWN>
V RM

r f BC-WHITE-HOUSE-STANDING  04-08 0112</UNKNOWN>
<TEXT>
<TITLE>WHITE HOUSE STANDING FIRM ON JAPANESE SANCTIONS</TITLE>
<DATELINE>  WASHINGTON, April 8 - </DATELINE><BODY>Presidential spokesman Marlin
Fitzwater said U.S. trade sanctions against Japan were likely
take effect on April 17 in spite of a "full court press" by
Japanese officials to avoid them.
    "All indications are they will take effect," he said.
    "I would say Japan is applying the full court press ... They
certainly are putting both feet forward in terms of explaining
their position," Fitzwater told reporters.
    He noted high level meetings on the trade dispute are
underway here but said, "I don't think there's anything I can
report and I don't believe there's been any official movement."
 reuterend
</BODY></TEXT>
</REUTERS>
```

Figure 4.5. A test document.

Test documents should also go through the process of parsing, tokenization, stop words removal and stemming. Significant terms are generated as shown in Figure 4.6.

appli
april
avoid
court
disput
don
effect
explain
feet
fitzwat
forward
full
high
indic
japan
japanes
level
marlin
meet
movement
note
offici
posit
presidenti
press
put
report
sanction
spite
spokesman
term
think
told
trade
underwai

Figure 4.6. A screenshot of significant terms in a test document

For each significant term generated determine whether the term occurs in the category or not, if it occurs then increment wf value. If it is a 1-itemset then wf equals 1, if 2-itemset wf equals 2 etc. In this way weights of all terms for each category is determined and which ever is having highest value the document is linked with that category. In this case when the test document 'd' is linked with the above five categories weight factors are:

wf of 'd' linked with Trade is 7

wf of 'd' linked with Grain is 2

wf of 'd' linked with Interest is 5

wf of 'd' linked with Acquisition is 1.

wf of 'd' linked with Jobs is 6.

Hence, given test document 'd' is mapped to category Trade. If sum of weight factors are equal for any two categories then it is the case that document d belongs to both the categories.

4.3 Precision and Recall

The performance of categorization model built is evaluated based on standard precision, recall and F1 values. Let TP be the number of true positives i.e. number of documents which both experts and the model agreed as belonging to the same category. Let FP be the number of false positives i.e. the number of documents that are wrongly categorized by the model as belonging to that category.

Precision is defined as:

$$precision = \frac{TP}{TP + FP}$$

Let FN be the number of false negatives, that is, the number of documents which are not labeled as belonging to the category but should have been.

Recall is defined as:

$$recall = \frac{TP}{TP + FN}$$

The harmonic mean of precision and recall is called the F1 measure is defined as [24]:

$$F1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

In this experiment by varying support threshold , θ precision, recall and F1 values are calculated.

4.4 Results

Table 4.7. Precision, recall and F1 values when θ = 5%.

Category	Total documents	TP	FP	FN	Precision	Recall	F1
Trade	58	54	6	4	0.90	0.93	0.92
Grain	4	4	0	0	1	1	1
Interest	56	55	6	1	0.90	0.98	0.94
Acquisition	70	63	0	7	1	0.90	0.95
Jobs	12	11	1	1	0.91	0.91	0.91

The average precision and recall values obtained are 94% and 95%

Table 4.8. Precision, recall and F1 values when θ = 10%.

Category	Total Documents	TP	FP	FN	Precision	Recall	F1
Trade	58	54	11	4	0.83	0.93	0.88
Grain	4	4	0	1	1	0.75	0.85
Interest	56	55	8	1	0.87	0.98	0.92
Acquisition	70	55	0	15	1	0.78	0.87
Jobs	12	11	1	1	0.91	0.91	0.91

The average precision and recall values obtained are 92% and 87%.

50

CHAPTER 5

RESULTS EVALUATION

This chapter evaluates the results presented in chapter 4. It clearly illustrates the reasons behind documents which are wrongly predicted by the model considering two categories: Acquisition and Jobs.

5.1 Evaluation for Category Acquisition

Out of 200 documents used for testing, Acquisition has 70 documents as shown in Table 4.6. When calculating precision and recall values it is observed that false negative value for Acquisition is seven as shown in Table 4.7. This means the model has predicted wrong category labels for seven documents out of 70 documents. By evaluation it is found that documents D_7, D_{23}, D_{25}, D_{51} are categorized to Trade and D_{22}, D_{33}, D_{43} to Interest instead of Acquisition as defined by experts. The reason is explained below by considering individual documents.

(i) Document D_7 is categorized to Trade instead of Acquisition because weight factor for D_7 linked with Trade is more than when it is linked with Acquisition as shown below.

D_7 with Acquisition:

Frequent 1-itemsets: approv, co, complet.

Weight factor for acquisition = 3

D_7 with Trade:

Frequent 1-itemsets: agreement, call, include, without.

Weight factor for Trade = 4.

Hence D_7 is categorized to Trade.

(ii) D_{23} is categorized to Trade instead of Acquisition because of more number of

1, 2-itemsets in D_{23} when it is linked with Trade

D_{23} with Acquisition:

Frequent 1-itemsets: agre, compani, complet, corp, plan.

Frequent 2-itemsets: agre compani, agreement compani, compani corp,

compani plan, part plan.

Weight factor for acquisition = 5 + 10 = 15.

D_{23} with Trade:

Frequent 1-itemsets: agreement, american, billion, negoti, problem, set,

sign, talk, time, told.

Frequent 2-itemsets: billion offici, billion plan, billion talk, billion told, offici talk,

offici time, plan told, plan week.

Weight factor for Trade = 10 + 16 = 26.

(iii) D_{25} is mapped with Trade because there are more 1-itemsets in D_{25} that

belong to category Trade than Acquisition.

D_{25} with Acquisition:

Frequent 1-itemsets: corp, group, manag, net, sale, sell, share.

Frequent 2-itemsets: corp share.

Weight factor for acquisition = 7 + 2 = 9.

D_{25} with Trade:

Frequent 1-itemsets: agreement, chairman, chief, full, include, industri, intern, negoti, unit.

Frequent 2-itemsets: include industry.

Weight factor for Trade = 9 + 2 = 11.

(iV) D_{51} is categorized as Trade because it has highest weight factor when compared with D_{51} linked with Acquisition.

D_{51} with Acquisition:

Frequent 1-itemsets: acquisit, bui, commiss, exchang, hold, sell, share.

Weight factor for acquisition = 7.

D_{51} with Trade:

Frequent 1-itemsets: drop, gener, include, partner, reduc, told.

Frequent 2-itemsets: include told.

Weight factor for Trade = 6 + 2 = 8.

(V) D_{22} is categorized by the model as belonging to Interest instead of Acquisition because there are more 2-itemsets in D_{22} that belong to Interest.

D_{22} with Acquisition:

Frequent 1-itemsets: bui, busi, co, compani, firm, held, manag, offer, plan, share, stock.

Frequent 2-itemsets: bui comapni, busi compani, compani offer, comapi plan, comapni share, compani stock, comapi share, share stock.

Weight factor for acquisition = 11 + 16 = 27.

D_{22} with Interest:

Frequent 1-itemsets: analyst, deal, debt, interest, invest, sourc, todai, week.

Frequent 2-itemsets: ad interest, billion debt, billion interest, billion spokesman, billion todai, billion week, interest month, interest plan, interest todai, interest week, intern week, month week, sourc week.

Weight factor for Interest = 8 + 26 = 34

(Vi) D_{33} is categorized under Interest because it has more frequent 2, 3-itemsets belonging to Interest rather than to Acquisition.

D_{33} with Acquisition:

Frequent 1-itemsets: acquisit, agre, bui, busi, capit, cash, co, compani, corp, make.

Frequent 2-itemsets: agre comapni, bui compani, busi compani, compani corp.

Weight factor for acquisition = 10 + 8 = 18.

D_{33} with Interest:

Frequent 1-itemsets: amount, analyst, expect, declin, fund, growth, low, secur.

Frequent 2-itemsets: billion expect, billion fund, billion secur, expect fund, expect secur, fund secur.

Frequent 3-itemsets: billion fund secur.

Weight factor for interest = 8 + 12 + 3 = 23

(Vii) D_{43} is mapped to Interest instead of acquisition as there are more 2-itemsets that belong to Interest.

D_{43} with Acquisition:

Frequent 1-itemsets: acquir, bui, comapni, frm, hold.

Frequent 2-itemsets: acquir compani, bui compani.

Weight factor for acquisition = 5 + 4 = 9.

D_{43} with Interest:

Frequent 1-itemsets: bank, interest, todai.

Frequent 2-itemsets: bank billion, bank foreign, bank interest, bank todai,

billion interest, billion todai, februari interest, foreign interest, interest todai.

Weight factor for Interest = 3 + 18 = 21.

5.2 Evaluation for Category Jobs

Category Jobs has 12 documents out of 200 test documents. From Table 4.7, it is observed that for category Jobs, false negative value is one i.e. one document in jobs collection is wrongly labeled. By evaluation it is found that document D_4 is wrongly categorized. It is mapped to category Trade instead of Jobs. The reason for this is there are only five terms in D_4 that belongs to 1-itemsets of Jobs but there are 15 terms in D_4 that belong to 1-itemsets of Trade including 33 2-itemset and two 3-itemsets. As explained in chapter 4, the model categorizes a new unseen document to only that category which has its sum of weight factor wf, to be maximum. This is illustrated below:

D_4 when mapped with category Jobs itemsets obtained are:

Frequent 1-itemsets: benefit, increas, manufactur, period, present.

Weight factor for jobs = 5.

D_4 when mapped with category Trade itemsets obtained are:

Frequent 1-itemsets: american, congress, countri, develop, export, gener, good, industri, issu, intern, product, state, tariff, told, unit.

Frequent 2-itemsets: countri export, countri increas, countri industri, countri product, countri state, countri told, countri unit, export gener, export good, export increas, export industri, export nation, export product, export state, export told, export unit, good state, increas industri, increas nation, increas product, increas state, increas told, industri nation, industri product, industri state, industri told, nation state, nation told, product state, product told, product unit, state told, state unit.

Frequent 3-itemsets: countri state unit, export state unit.

Weight factor for trade = 15 + 66 + 6 = 87

Hence document D_4 is labeled as belonging to category Trade.

The false positive value for Jobs is also one i.e. some other category document is labeled as belonging to this category. It is found that in category Trade document D_{10} is categorized as belonging to Jobs because there are more 2-itemsets in D_{10} that belong to Jobs as shown below:

When D_{10} is linked with Trade, itemsets obtained are:

Frequent 1-itemsets: export, import, surplus.

Frequent 2-itmesets: export figur, export import, export januari, export month, export surplu, import januari, import surplu.

Weight factor for Trade = 3 + 14 = 17.

When D_{10} linked with Jobs, itemsets obtained are:

Frequent 1-itmesets: februari, fell, figur, januari, previou, record, total.

Frequent 2-itemsets: februari fell, februari janauari, februari month, februari total, fell januari, fell total, figur januari, figur total, januari month, janauari total, month total.

Frequent 3-itemset: februari januari month.

Weight factor for Jobs = 7 + 22 + 3 = 32.

By observing all the above evaluation it is clear that cardinality of itemsets plays an important role in making the model predict category labels for new documents.

CHAPTER 6

CONCLUSION AND FUTURE WORK

The premise of this thesis was to come to a decision on the model built using frequent itemsets i.e. Can frequent itemsets be efficiently used to perform text categorization? After doing this experiment on a collection of Reuters 21578 test documents, the precision and recall values obtained with varying thresholds as shown in chapter 4 conclude that this is an efficient model to perform automatic document categorization. User-defined threshold value plays an important role in deciding whether an itemset is frequent or not. After evaluating the results in chapter 5, it can be concluded that cardinality of itemsets is important to a model in deciding whether a document belongs to a particular category or not.

Text categorization is an active research area in information retrieval and machine learning. This work can be extended by training and testing the model built, on large document collections determining their precision and recall values. Also, this model can be compared with various text categorization models available and determine which model performs better in a commercial environment.

.

BIBLIOGRAPHY

1.   Fabrizio Sebastiani, 'Machine Learning in Automated Text Categorization',Italy,2002.
      http://nmis.isti.cnr.it/sebastiani/Publications/ACMCS02.pdf.

2.   Fabrizio Sebastiani, ' Text Categorization', University of Padova, Italy, 2005.
      http://nmis.isti.cnr.it/sebastiani/Publications/TM05.pdf.

3.   Ricardo Baeza Yates, Berthier Riberio Neto, 'Modern Information Retrieval', Addison Wesley, Chapter 1, Addison Wesley Longman,1999.
      http://people.ischool.berkeley.edu/~hearst/irbook/1/node2.html.

4.   Margaret H. Dunham, 'Data Mining Introductory and Advanced Topics', Chapter 1, 2 and 4, Southern Methodist University, Pearson Education Inc, 2003.

5.   Wikipedia, the free Encyclopedia, Data Mining, 2008.
      http://en.wikipedia.org/wiki/Data_mining

6.   Machine Learning. The Free On-line Dictionary of computing. Retrieved November 24, 2008, from the Dictionary.com website:
      http://dictionary.reference.com/search?q=machine+learning

7.   American Association for Artificial Intelligence (AAAI) Inc., A Nonprofit California Corporation, AI Topics / Machine Learning, 2008.
      http://www.aaai.org/AITopics/pmwiki/pmwiki.php/AITopics/MachineLearning

8.   Wikipedia, the free Encyclopedia, Machine Learning, 2008.
      http://en.wikipedia.org/wiki/Machine_learning

9.   Jiawei Han and Micheline Kamber, 'Data Mining concepts and Techniques', Chapter 7, Simon Fraser University, Morgan Kaufmann Publishers, 2001.

10.   Pang-Ning Tan, Michael Steinbach, Vipin Kumar, 'Introduction to Data mining', Chapters 1, 5, 6, Pearson Addison Wesley, 2005.

11. Rakesh Agarwal and Ramakrishnan Srikant, 'Fast Algorithms for Mining Association Rules' pages 580-592 from Michael Stonebraker, Joseph M. Hellerstein 'Readings in database systems', Third Edition, Morgan kaufmann Publishers,1998.

12. James Joyce, 'Bayes Theorem' Stanford Encyclopedia of Philosophy, June 2003.
http://plato.stanford.edu/entries/bayes-theorem

13. Wikipedia, the free Encyclopedia, Naive Bayes Classifier, 2008.
http://en.wikipedia.org/wiki/Naive_Bayesian_classification

14. Frank Keller, 'Naive Bayes classifiers- connectionist and statistical languageProcessing'(n.d.)
http://homepages.inf.ed.ac.uk/keller/teaching/connectionism/lecture10_4 up.pdf.

15. Howard Hamilton, Ergun Gurak, Leah Findlater, and Wayne Olive, 'Apriori Itemsets Generation' from ' Knowledge Discovery in Databases', 2003.
http://www2.cs.uregina.ca/~dbd/cs831/notes/itemsets/itemset_apriori.html.

16. Pang-Ning Tan, Michael Steinbach, Vipin kumar, 'Introduction to Data Minig', figures from this website (n.d.):
http://www-users.cs.umn.edu/~kumar/dmbook/index.php

17. Jiri Hynek, Karel Jezek, Ondrej Rohlik, 'Short Document Categorization – Itemsets Method ', ERIC Laboratories, 2000.
http://eric.univ-lyon2.fr/~pkdd2000/Download/WS4_02.pdf

18. David D. Lewis , ' Reuters 21578, Distribution 1.0 Test collection' (n.d.)
http://www.daviddlewis.com/resources/testcollections/reuters21578/

19. Dr. E. Garcia, 'Document Indexing Tutorial', 2006.
http://www.miislita.com/information-retrieval-tutorial/indexing.html

20. Kiran Pai, 'A simple way to read an XML file in Java', 2002.
http://www.developerfusion.com/code/2064/a-simple-way-to-read-an-xml file-in-java/

21. HappyCoders,'TokenizingJavasourcecode'(n.d.)
http://www.java.happycodings.com/Core_Java/code84.html

22. Martin Porter, 'The Porter Stemming Algorithm', Jan 2006.
http://tartarus.org/~martin/PorterStemmer/.

23. Wikipedia, the free Encyclopedia, Inverted Index, 2008. http://en.wikipedia.org/wiki/Inverted_index

24. Yiming Yang, Jan O. Pederson, 'A comparative study on feature selection in text categorization', Proceedings of the fourteenth international conference on machine learning, pages: 412-420, 1997.

25. Kazem Taghva, Jeffrey Coombs, Ray Pereda, Thomas Nartker, 'Address Extraction Using Hidden Markov Models', Information Science Research Institute, UNLV where precision and recall definitions are taken from this website: http://www.isri.unlv.edu/publications/isripub/Taghva2005a.pdf

VITA

Graduate College
University of Nevada, Las Vegas

Prathima Madadi

Local Address:
    2255 E Sunset Road,
    #2024,
    Las Vegas, NV - 89119

Home Address:
    7574 Erinway,
    Cupertino, CA - 95014

Degrees:
    Bachelor of Technology in Computer Science and Engineering, 2006
    Jawaharlal Nehru Technological University, India

Thesis Title: Text Categorization Based on Apriori Algorithm's Frequent Itemsets

Thesis Examination Committee:
    Chairperson, Dr. Kazem Taghva, Ph.D.
    Committee Member, Dr. Ajoy K. Datta, Ph.D.
    Committee Member, Dr. Laxmi P. Gewali, Ph.D
    Graduate College Representative, Dr. Muthukumar Venkatesan, Ph.D.