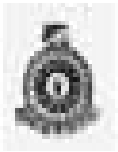
# Automatic Word Clustering

# in

# Application of Open-Ended Response Categorization

**N P K Medagoda**

**December 2012**



# Automatic Word Clustering in Application of Open-Ended Response Categorization

**A thesis submitted for the Degree of Master of Philosophy**

**N.P.K. Medagoda**
**University of Colombo School of Computing**
**December 2012**

The Thesis is my original work and has not been submitted previously for a degree at this or any other university/institute. To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Author's name N.P.K. Medagoda Date ……

Signature …….

This is to certify that this thesis is based on the work of Mr. N.P.K. Medagoda under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by

Supervisor Name Dr. Ruvan Weerasinghe Date ……...

Signature ……..

# Abstract

Open ended questions are an essential and important part of survey questionnaires. They provide an opportunity for researchers to discover unanticipated information regarding the domain of study. However, they are problematic for processing since they are unstructured questions to which possible answers are not suggested, and the respondent is free to answer in his or her own words. This thesis presents novel methods of categorizing such open ended survey responses**.** A document clustering technique is employed in this study to categorize responses to open-ended survey questions. Supervised and unsupervised methods of categorizing open ended responses are tested in the study.

Initially the author proposed a hierarchical clustering based algorithm as the unsupervised method to code the open-ended responses which were not labelled at all. The algorithm employs several natural language processing techniques to extract a classification of responses automatically. Naive Bayes classification was proposed as the supervised solution. This Naive Bayes algorithm was proposed for the open ended responses which were partially labelled.

Two experiments were carried out to determine the accuracy of the proposed algorithms which proved to be promising. Hierarchical clustering based algorithm shows more than 70% accuracy when compared with the manually coded responses. The proposed Naive Bayes algorithm didn't not illustrate the results as it expected. Therefore Positive Naive Bayes algorithm was introduced and it achieved an overall performance of 80%

# Contents

List of Figures

List of Tables

# Acknowledgement

I wish to express my sincere thanks to my supervisor Dr. Ruvan Weerasinghe, Senior lecturer University of Colombo School of Computing, for the invaluable knowledge and the important guidance extended to achieve the goals of this project and for the unstinting support given throughout my career.

Furthermore, I thank all the past and present academic members and other staff of the Language Technology Research Laboratory, at the University of Colombo School Of Computing for the support extended to me.

Next, special thanks go to my wife, daughter, son and rest of the family members for the inspiration given and the sacrifices they have underdone during the period of my MPhil Research.

# Chapter 1: Introduction, Motivation and Goals

## 1.1 Chapter overview

This chapter explains the main objective of the research, motivation behind the study and goals of the dissertation work. The chapter begins with a detailed introduction of the study in the domain of open-ended response categorization. Motivation of the research was presented in the chapter following the Goals and objectives of the study. The structure of the thesis is explained at the end of the chapter.

## 1.2 Introduction

Statistical surveys are the systematic methods of collecting information in many situations in today's world. The main tool of the statistical survey is the questionnaire. When building a questionnaire for a statistical survey, it is essential to include open-ended questions to gather unanticipated information. Open-ended questions are those questions that will elicit additional information from the respondents. Open ended questions in survey questionnaires are unstructured questions in which possible answers are not suggested, and the respondent is expected to answer it in his or her own words. Since the freedom of answering these types of questions is given to the respondent, the respondent can write any answers which are related to the question. Such WH-questions usually begin with "how", "what", "when", "where", or "why". Therefore it is clear that there is no specific format to answer these questions. In analyzing such responses we need to filter appropriate sentences, words from the responses. Often however, the responses to this type of question in surveys are neglected owing to the difficulty in classifying them into any useful form.

The main advantage of including these types of questions is getting more information than the 'closed questions'. The complexity of data collection, the unpredictable space for storage of responses and the greater time needed for analysis are the main difficulties of dealing with open ended questions in a survey (Bullington et al., 1998).

In paper based surveys, open ended responses are categorized using code books. These pre-defined codes are available before the survey is conducted or sometimes after conducting the pilot survey for a small sample. It has been identified that these codes are domain specific and some codes do

not correspond to the responses of the respondents. Only experienced domain specific experts can compile these codes, and finding such personnel presents a severe bottleneck in the industry. This kind of human intensive process is also time consuming and the resulting accuracy is usually very low (Giorgetti and Sebastiani, 2003).

## 1.3 Motivation

More information is available at our finger tips than ever before. The ability to access information has grown tremendously with the introduction of World Wide Web in the mid 1990s. We can access and find any information about any topic using millions of pages available on the web. The addition of more and more pages to the web is increasing rapidly not just by megabytes but by terabytes. The web is not only to store and access information but for many other online processes as well.

Unlike in the past, today many statistical surveys are conducted over the internet for speedy processing and higher accuracy. Paper based surveys lead to high cost of administration and excessive time taken to complete the survey. As a result of these disadvantages, many researchers now tend to carry out their surveys on-line. The widely available online panels motivate the researcher to carry out online surveys especially in market research, product evaluation, and service evaluation among others.

Information is of no use unless we can retrieve the correct information and analyze within few seconds. The concept of text analysis has been introduced in order to overcome the problem of analyzing and retrieving the correct information.

Since the responses to the open-ended question are available electronically, these text processing techniques can also be applied very easily without any limitations. Responses to the open-ended questions include cognitive aspects such as opinions, emotions and sentiments. Since the approaches of natural language processing are common in text processing some cognitive aspects included in the answers can be extracted using these techniques.

To solve the problem of analyzing open-ended responses we are motivated to investigate the application of text processing techniques in open-ended responses. In categorizing open-ended responses text processing methods are proposed in the algorithm.

## 1.4 Goals of the Research

The main goal of the research is to build an algorithm to categorize the open ended responses. Understanding the uses of open ended questions and open ended responses in statistical surveys is essential in this study. Knowledge of the available open ended response categorization is a sub goal in this research. Comprehensive literature review to gain the knowledge of available categorization techniques is to be carried out in the proposed study.

There are many objectives related to the study. Preprocessing is the vital task in any text processing work. In this research comprehensive data cleaning and normalizing techniques are supposed to be employed as the preprocessing step of the algorithm. Therefore finding the preprocessing techniques used in information extraction and text mining applications, is one major objective. Secondly understanding the benefit and uses of world clusters in open ended responses was set as an objective. The proposed algorithm can only use the key word contained in the open ended responses, find methods for extracting such a key word list is a significant objective of the study. The feature vector is to be constructed as the input to the hierarchical clustering algorithm. Another objective was set to find the techniques of building such vectors. Supervised method of categorizing open ended responses is proposed in this study. An objective is set to research the supervised methods applicable in text categorization algorithms.

## 1.5 Scope of the Research

The scope of the study is limited to the open ended responses written in English language. Unavailability of open ended responses electronically typed in other languages and the lack of resources such as stemming algorithms are the main constraints considered for the selection. It is aimed to categorize the open ended responses which are electronically available. Responses containing less than 10 sentences were considered as the single open ended response. That is responses with many paragraphs were omitted in the study.

## 1.6 Structure of the dissertation

The rest of this thesis is organized as follows:

The next chapter explains the findings and the analysis of open-ended questions and their responses in the literature. This explains the types of open-ended questions asked in statistical surveys and how the respondents typically answer these questions.

The third chapter describes the method of obtaining word clusters and uses of these clusters in text analysis research. In most of the text analysis applications the basic requirement is retrieving the word clusters for further analysis.

Chapter four discussed the techniques of text classification in detail. Supervised and Unsupervised methods used in the classification of text analysis are explained with underlying theoretical frame work. Next chapter describes the proposed hierarchical clustering algorithm with the sequence of tasks to be carried out. The supervised classification algorithm for open-ended response coding is presented in chapter six. The experiment done to prove the hierarchical clustering algorithm has described in chapter seven with the evaluation and accuracy measurements. Proving the appropriateness of the supervised classification algorithm is discussed with a real application and explained in chapter eight. Finally the discussion and conclusion with future work is explained in chapter nine.

# Chapter 2: A Survey on Open-Ended Questions & Responses

## 2.1 Chapter Objective:

This chapter describes the profile of Open-Ended questions, Responses to the open-ended questions and the importance of Open-Ended questions in statistical surveys. The literature reviews of open-ended response analyzing techniques are presented in the latter part of the chapter.

## 2.2 Open-Ended Questions & Responses

Questionnaire is the primary tool of gathering information for statistical surveys. A well defined and logically ordered questionnaire facilitates much to bring the correct and timely data to the study. There are many classifications of questions that are included in a statistical survey. The most popular classification is close ended and open ended questions. The respondents answers are limited to a fixed set of responses known as closed ended questions. Yes/No, Multiple choice and scaled questions are some example of closed ended questions. Open ended questions are unstructured questions in which possible answers are not suggested, and the respondent answers it in his or her own words. No predefined categories or responses are suggested in this type of questions. When building a questionnaire for a statistical survey, it is essential to include open-ended questions in questionnaires to gather more information. Open-ended questions are those questions that will inquire additional information by the responses. Since the freedom to answering these types of questions, the respondent can write any answers which are related to the question. Such questions usually begin with a “how, what, when, where, and why”. Therefore it is clear; there is no specific format for these answers.

The method of presenting the open ended questions to the respondent is classified as; completely unstructured, sentence completion, story completion and picture completion. A common feature of all this type of questions is, answers by the respondents are differing in each record.

Even though the responses are varying in open ended questions there is a hidden and common opinion that can be identified by grouping the similar responses together. Therefore the first step

of analyzing open ended responses is grouping or clustering the responses by considering their similarities.

In analyzing such responses we need to filter appropriate sentences, words from the responses.

The main advantage of including these types of questions is getting more information than the close ends. Complexity of data collection, more spaces for storage and taking more time for analysis are the main difficulties of including open ended questions in a survey.

A detailed survey is carried out for in depth investigation on open ends. As mentioned earlier most of the open-ended questions start with "what, how and why" functions words. Whereas some begins with "please type, please list" phrases. The responses can be long sentences, multiple words or a single word. The response rate to the open ended questions varies from 30% to 90%. Following are some of the findings.

TABLE 2.1: Open ended Questions and Responses

| **Survey** | **Question** | **Number of Respondents** | **Number of Responses** | **Attributes of the Responses** |
|---|---|---|---|---|
| Survey 1 | *How do you think the roles of The Energy Savings Trust and The Carbon Trust differ?* | 158 | 155 | Long Sentences, with more than one sentence per response, Single word, Multiple words |
| Survey 2 | *What else could encourage you and your organization to contact and engage with such a scheme, with a view to reducing your carbon emissions?* | 158 | 152 | Long Sentences, with more than one sentence per response,, Single word, Multiple words |
| Survey 3 | *What could discourage you and your organization?* | 87 | 82 | Long Sentences, Single word, Multiple words |
| Survey 4 | *Please type in everything you remember about the last advertisement you saw/heard for McDonald's, including what it said and what it showed.* | 175 | 173 | Long Sentences, Single word, Multiple words |
| Survey 5 | *Please type in any items that you know are currently available on the McDonald's 'Pound Saver Menu'* | 286 | 282 | No sentences, Single word, Multiple words |

| | | | | |
|---|---|---|---|---|
| Survey 6 | *Please list any websites where you have seen advertising for the McDonald's Pound Saver Menu.* | 53 | 51 | No sentences, Single word, Multiple words |
| Survey 7 | *What could be done to improve your satisfaction with this mobile phone?* | 1800 | 624 | Sentences with maximum 10 words, Single word, Multiple words |
| Survey 8 | *What could this network do to improve your satisfaction?* | 1800 | 507 | Sentences with maximum 8 words, Single word, Multiple words |

## 2.3 Methods of Analyzing Open-Ended Responses

Extracting hidden information contained in the open ended responses and applying the statistical techniques on extracted information can be defined as analyzing open ended questions. Analyzing the open ended questions begin with categorizing the responses. Currently, in most of the statistical surveys this is done by visual inspection by the researcher. The issue of the visual inspection is time consuming and need several humans to carry out. Due to human involvement the categories can be different to each researcher. Once the categories are defined the next step is to label the responses with one or several categories. The process is known as "coding". Again a manual process is applied to this step and the decision depends highly on human intervention.

The literature available for automatic open ended response categorizing is limited and some previous attempts reveal that the majority of them are analyzing a particular study.

A study of "Automated text clustering on responses to open-ended questions" was carried out by M. Kang et al. (2005) and the study was for the purpose of course evaluation. A new method has been introduced to measure the similarity of responses. This method emphasizes word synonymy to improve the scantiness. The word synonymy is measured by the distance between two terms. The distance is defined by using the concept calculation dictionary. The feature vector consists of term frequency related to each synonymy group in the union set of two terms. The clustering of the feature vectors has been done by using the hierarchical clustering methods. The average accuracy is around 73% and it's stated that the classification accuracy is affected the by the dictionary accuracy.

Another study of "Method of Atypical Opinion Extraction from Answers in Open-ended Questions" done by Ayako Hiramatsu et al. (2005) and the responses written in Japanese language are from the open ended questions supplied by users when they unsubscribed the service of mobile games. The proposed system in this study includes classifying the opinion as typical or atypical and divides the opinion into word phrases in each typical word combination. Opinions having the same meaning as items of closed ended questions, frequent opinions and irrelevant opinions are defined as typical opinions. Whereas, not typical responses are the "atypical" opinions. They have defined and maintained a database of typical opinions which contains the words like high, expensive etc. A machine engine to extract atypical opinions compares key words of the given opinions with the typical words in database. There are several comparison methods defined in the study and state the problem of these comparing methods. This method has a weakness that long sentence containing more than one opinion is wrongly classified as typical even though it includes atypical opinion. Application experiment shows less satisfactory results since opinions with short sentences having 3 or 4 key words cause the low recall score.

The study of "Multiclass Text Categorization for Automated survey Coding" is completed by D. Giorgetti and F. Sebastiani (2003) to code the open-ended responses using supervised learning techniques. They formulated the problem of survey coding as a multiclass text categorization problem. Hence, authors present two different learning techniques to categorize the survey responses. One method based on Naïve Bayesian classification and another based on multiclass Support Vector Machine (SVM). In Bayesian learning the feature vector is constructed using tf*idf (term frequency * Inverse document frequency) of the key words extracted from the responses. An experiment was carried out for a response set with 7 predefined categories, using the proposed algorithms. The results reveal that for some codes Bayesian classification shows higher accuracy than the SVM approach. Results also were compared with results of dictionary based approach. The accuracy of proposed methods is much higher than the dictionary methods. In above method the number of codes is limited to a few codes.

In this study the author is proposing a generic system of coding the open-ended responses for any domain. In proposed system the number of codes is not limited and predefined. Hence hierarchical clustering algorithm is the appropriate clustering algorithm. The user can define or

chose the number of categories by considering the hierarchy of the clusters. Method described automating the coding step and hence reduces the time complexity and human involvements. A meaningful topic for the grouped responses is suggested by the proposed method and that can be used for future responses.

# Chapter 3: Word Clusters in Open-Ended Response Categorization

## 3.1 Chapter Objective:

This chapter explains the importance of word clusters in text analysis application such as open-ended response analysis. Chapter begins with the discussion of taxonomy of word similarities. Techniques used to construct the word clusters are presented in detail at middle of the chapter. Mining the features for constructing the vectors for text analysis and the methods of text clustering were discussed in proceeding sections of the chapter.

## 3.2 Word clustering

As explained in chapter 1 "word clusters" are the main ingredients in open-ended response categorization and analysis. Most words in natural languages have multiple possible meanings that can only be determined by considering the context in which they occur. Given a target word used in a number of different contexts, word clustering is the process of grouping these instances of the target words together by determining which contexts are the most similar to each other. This is motivated by Miller and Charles (1991), who hypothesize that words with similar meanings are often used in similar contexts. Hence, word sense discrimination reduces the problem of finding classes of similar contexts such that each class represents a single word sense. Word clustering is a technique for partitioning sets of words into subset of semantically or syntactically similar words. The problem of word clustering is also refers as constructing a thesaurus based on the similarity of the words.

The similarities of words are two types [WWW 01]

- **Paradigmatic Similarity**

This is also known as substitutional similarity. Two words that are paradigmatic similar may be substitute for one another in particular text

- **Syntagmatic Similarity**

Two words that are syntamatically similar means significantly occur together in two texts. For instance read and book are syntamatically similar since they co-occur within the same context.

## 3.3 Application of word clustering

Word clustering is increasingly becoming a major technique used in a number of natural language processing tasks varying from word sense or structural disambiguation to information retrieval and filtering. One of the most frequent application domains is text or document classification.

## 3.4 Types of Clustering Algorithms

There are two types of structures provided by clustering algorithms, Hierarchical and Non - Hierarchical clustering.

Non – Hierarchical clustering simply consist of a certain number of clusters and the relation between them is undetermined. They start with a set of initial clusters and improve them by iterating a reallocation operation that reassigns objects.

A hierarchical clustering is a hierarchy with the usual interpretation that each node stands for a subclass of its mother's node. In hierarchical clustering assignment is usually hard. In hard assignment each object is assigned to one and only one cluster.

In non-hierarchical clustering both soft and hard assignment are common. Soft assignments allow degrees of membership and membership in multiple clusters. The difference from hard clustering is that there is uncertainty about which cluster is the correct one. A hard clustering algorithm has to choose one cluster to which to assign every item. It is common place that many words have more than one part of speech. The best that can be in such cases is to define additional clusters corresponding to word that can cause confusion. Soft clustering is therefore somewhat more appropriate for many problems in natural language processing.

## 3.5 Hierarchical clustering Techniques

a. Agglomerative Clustering
b. Divisive

### 3.5.1 Agglomerative Clustering

Agglomerative clustering starts with single object that is; initially they are many clusters as objects. It is common practice to begin with each observation in a cluster by itself. The most similar objects are first grouped and these initial groups are merged according to their similarities. Finally, as the similarity decreases all subgroups are fused in to a single cluster (Jain A.K, et al. 1999)

The most practical agglomerative hierarchical procedure is "linkage methods". The linkage methods are suitable for clustering observations as well as variables. The choice of which clusters to merge or split is determined by a linkage criterion. Linkage is a function of pair wise distance between objects. A non negative valued function may be used as a measure of similarity between pairs of observations.

The result of linkage clustering can be graphically displayed in the form of a dendrogram , or a tree diagram. The branches of the tree represent the clusters. The branches come together at nodes whose positions along a similarity distance axis indicate the level at which the fusion occurs.

In this study the author is discussing three linkage methods. The single linkage (minimum distance), complete linkage (maximum distance) and average linkage (average distance), (Johnson R.A, Wichern D. E, 1996)

- **Single Linkage**

The inputs of a single linkage algorithm can be the distance or similarities between pairs of objects. Groupes are formed by the individual entities by merging nearest neighbors, where the term nearest neighbor means smallest or largest similarity.

Initially find the smallest distance $D = \{d_{ik}\}$ and merge the corresponding objects. Say U and V, to get the cluster (UV)

Update the entities of the distance matrix by deleting the rows and columns corresponding to U and V and adding a row and column giving the distance between clusters (UV). The distance between cluster UV and any other cluster W are computed by

$$d_{(uv)w} = \min\{d_{uw} , d_{vw}\}$$

The tendency of single-link clustering to produce elongated clusters is some times called chaining effects. This is due to the large similarities without taking into account the global context.

- **Complete Linkage**

This proceeds much the same way as single linkage, with one important exception. The distance between clusters is determined by the distance between two elements, one from each cluster that is most distant.

In this case the distance between cluster UV and any other cluster W are computed by

$$d_{(uv)w} = \max\{d_{uw} , d_{vw}\}$$

Complete linkage clustering has a similarity function that focuses on global cluster quality.

- **Average Linkage**

Average linkage treats the distance between two clusters as the average distance between all pairs of items where one member of a pair belongs to each cluster. Hence the distance between cluster UV and any other cluster W are computed by

$$d_{(UV)W} = \frac{\sum_i \sum_k d_{ik}}{N_{(UV)} N_W}$$

Where dik is the distance between object i in the cluster (UV) and object k in the cluster W and N(UV) and NW are the number of items in cluster (UV) and W respectively.

### 3.5.2 Divisive

Top-down or bottom-up is taxonomy of hierarchical clustering algorithm. Bottom-up is algorithms treat each document as a singleton cluster at the outset and then successively merge and known as Agglomerative clustering. In top-down clustering starts at the top with all documents in one cluster. The cluster is split using a flat clustering algorithm. This procedure is applied recursively until each document is in its own singleton cluster. Top-down clustering is conceptually more complex than bottom-up clustering since it needs a second, flat clustering algorithm and there is evidence that divisive algorithms produce more accurate hierarchies than bottom-up algorithms in some circumstances.

### 3.5.3 Similarity Measures in Hierarchical Clustering

To merge (Agglomerative) or divide (Divisive) groups which are similar or apart is done by the distance between the groups. Such a distance is known as the similarity measure of the cluster. There are several similarity measures used in hierarchical clustering. Euclidean distance, Manhattan distance, Squared Euclidean distance and Cosine similarity are among the most widely used. Let a and b are two vectors corresponding to two records which are going to cluster, then some of the applicable similarity measures in text processing are defined as (Johnson and Wichern, 1996),

a. Euclidean distance, $\|a - b\| = \sqrt{\sum_i (a_i - b_i)^2}$
b. Squared Euclidean distance, $\|a - b\|^2 = \sum_i (a_i - b_i)^2$
c. Manhattan distance, $\|a - b\| = \sum_i |a_i - b_i|$
d. Cosine similarity, $\frac{a.b}{|a|.|b|}$

## 3.6 Non-Hierarchical Clustering

In non-hierarchical clustering, objects are placed in only one cluster, and the relation between clusters is not really important. This method is also classified in hard clustering group. In hard assignment objects are put into one cluster and only one cluster. Method start with initial (random) set of clusters and then reallocates the objects to the currently best cluster by several passes. The clustering process is stopped by the measure of goodness or cluster quality. There are several techniques for hard clustering. The simplest algorithm is K-means.

### 3.6.1 K-means clustering

In K-means, it is known beforehand how many clusters there are supposed to be. Furthermore, each data element is represented by a set, or vector, of features. These features could be the word count of the document, the x and y coordinates of some point, etc. The centroid of each cluster is a vector of an equal number of features as the data elements that represents the average of the data elements within the cluster. The centroids are initially set randomly (though we diverge from this), and after they are set each data element is assigned to the cluster with the closest centroid. The centroids of each cluster are then recalculated as the average of all of the

data elements within the cluster, and the data elements are afterwards reassigned. This iteration continues until the clusters stop changing (Johnson and Wichern, 1996).

### 3.6.2 Expectation-Maximization (EM)-algorithm

The other basic non-hierarchical clustering technique is the uses of Expectation-Maximization (EM) algorithm. Here view each cluster as a Gaussian mixture (a normal distribution, bell curve), or as cause that helps create the underlying data. In EM algorithm estimates the parameters for Gaussion mixture.

Let $\theta_j = (\vec{\mu}_j, \Sigma_j, \prod_j)$ be the distribution of gaussion mixture model and the parameters of the $\Theta = (\theta_1, \ldots\ldots\ldots\ldots \theta_k)$ model,

The log likelihood of the data X given the parameters Θ is

$$l(X|\Theta) = \log \prod_{i=1}^{n} P(\vec{x}_i) = \log \prod_{i=1}^{n} \sum_{j=1}^{k} \pi_j n_j(\vec{x}_i; \vec{\mu}_j, \Sigma_j)$$

$$= \sum_{i=1}^{n} \log \sum_{j=1}^{k} \pi_j n_j(\vec{x}_i; \vec{\mu}_j, \Sigma_j)$$

The set of parameters Θ with the minimum likelihood gives us best model of the data. The EM algorithm is an iterative solution to the following circular statements:

**Expectation**: If we knew the value of Θ we could compute the expectation of the hidden variables (e.g., probability of x belonging to some cluster)

**Maximization**: If we knew the expected value hidden structure, we could compute the maximum likelihood value of Θ

### 3.7 Word Clusters using Language Model

In statistical language modeling, one technique to reduce the problematic effects of data sparsity is to partition the vocabulary into equivalence classes. Class-based n-gram models are intended to help overcome this data sparsity problem by grouping words into equivalence classes rather

than treating them as distinct words and thus reducing the number of parameters of the model (Brown et al., 1990).

A language model assigns a probability to a piece of unseen text, based on some training data. Statistical language model has been used for various natural language processing tasks including speech recognition, word clustering and Information retrieval.

Language models are generative models, which are models that define a probability mechanism for generating language. Such generative models might be explained by the probability mechanism. This mechanism defines a probability selecting a term T from a given document D as P(T |D). Suppose the process is repeated n times, selecting one at a time the terms $T_1$, $T_2$,. . . , $T_n$. Then, assuming independence between the successive events, the probability of the terms given the document D is defined as follows:

$$P(T_1, T_2, \ldots\ldots\ldots T_n) = \prod_{i}^{n} P(T_i|D)$$

The main notion of these models is that the probability of a word depends only upon the last N-1 words, the number of parameters in N-gram models increases considerably as N increases, resulting in an increase in the size of the model and the data required for training. Therefore, low values of N, usually 2-4 are employed causing the loss of long-term context information (Manning and Schutze, 1999)

Given sequence of M words W = $w_1$, $w_2$,…….$w_M$ a Language model estimates the priori probability P(W) = P($w_1$, $w_2$,…….$w_M$)

### 3.7.1 N-gram Model

In this traditional stochastic language model, the current word is predicted based on the preceding word (bigram) or the preceding N-1 words (N-gram) expecting that most of the relevant syntactic information lies in the immediate past.

It is a priori probability P(W) is expressed using conditional probabilities by

$P(W) = P(w_1)\prod P(w_i/w_1\ldots\ldots w_{i-1})$

For simplicity reasons by invoking the Markov chain assumptions, P($w_i$/$w_1$……$w_{i-1}$) is approximated by P($w_i$/$w_{i-N+1}$……$w_{i-1}$) with N= 2,3 or 4 as the expense of preserving the syntactic and semantic information from the more distant words.

### 3.7.2 Entropy

The criterion that is used in the language model is optimizing the decrease in cross entropy or perplexity, the amount by which the language model reduces the uncertainty about the next word.

Aim is to find a function π that assigns words to clusters which decreases perplexity compared to a N-gram model. (Manning and Schutze, 1999)

First approximate the cross entropy of the corpus

L = $w_1$, $w_2$ ….$w_M$ for π by making Markov assumption.

$$H(L,\pi) = -\frac{1}{M}\log P(w_1 \ldots\ldots w_M)$$

$$H(L,\pi) \approx -\frac{1}{M-1}\log \prod_{i=2}^{M} p(w_i | w_{i-1})$$

This can be simplified

$$H(L,\pi) = -\sum_{w} P(w)\log P(w) + \sum P(c_1,c_2)\log\frac{P(c_1,c_2)}{P(c_1)P(c_2)}$$

### 3.7.3 Mutual Information

$$H(L,\pi) = H(w) - I(c_1;c_2)$$

So we can minimize the cross entropy by choosing the cluster assignment function π such that the mutual information $I(c_1,c_2)$ between adjacent clusters is maximized.

In each step of clustering we select two clusters whose merge causes the smallest loss in mutual information.

$$MI - loss(c_i;c_j) = \sum I(c_k;c_i) + I(c_k;c_j) - I(c_k;c_i \cup c_j)$$

Then the selection of pairs of clusters that has to be merged according to the following equation

$$(c_{n_1}, c_{n_2}) = \arg\min_{(c_i,c_j)\in C\times C} MI - loss(c_i,c_j)$$

# Chapter 4: Text Categorization Techniques for Open-Ended Responses

## 4.1. Chapter Objective

The techniques use in text categorization those applicable for open-ended responses coding are presented in this chapter with their theoretical framework. The discussion begins with the introduction and significance of text categorization in information retrieval. The high level classification of the methods such as Supervised and Unsupervised techniques were presented in detail. The main techniques use in the proposed algorithm has been comprehensively explained in the following paragraphs.

## 4.2. Introduction

Text Categorization is very broad and active area of information research. Categorization defines as act of sorting and organizing things according to group, class, or, as you might expect, category (WWW 02). Text categorization is the task of automatically building automatic categories, by means of machine learning techniques. The domain of text categorization can be a set of words, sets of lines, sets of paragraphs or even sets of documents. The particular domain is selected on the requirement of the classification. With the rapid growth of online information the document categorization has became one of the key techniques for handling and organizing online text data.

Automatic classification of documents is an increasingly important tool for handling millions of documents in World Wide Web. Today millions of documents are accumulating in the internet. Hence the ability of retrieving a correct document is much more apparent. Therefore developing a more efficient and effective user friendly tool for retrieving correct information has great demand in the cyber world.

One of the reasons to build taxonomy of documents is to make it easier to find relevant document.

## 4.3. Classification of Text Categorization Techniques

Classification is a Machine Learning (ML) technique used to predict group membership for data instances. Every instance in any dataset used by machine learning algorithms is represented using the same set of features. The features may be continuous, categorical or binary. If instances are given with known labels then the learning is called supervised in contrast to unsupervised learning, where instances are unlabeled. By applying these unsupervised algorithms, researchers hope to discover unknown, but useful, classes of items. Therefore the main taxonomy of the classifications techniques are Supervised and Unsupervised classification.

In supervised algorithms, the classes are predetermined. These classes can be conceived of as a finite set, previously arrived at by a human. In practice, a certain segment of data will be labeled with these classifications. The machine learner's task is to search for patterns and construct mathematical models. These models are then evaluated on the basis of their predictive capacity in relation to measures of variance in the data itself. Decision tree, naive Bayes, etc are examples of supervised learning techniques.

Unsupervised learners are not provided with classifications. In fact, the basic task of unsupervised learning is to develop classification labels automatically. Unsupervised algorithms seek out similarity between pieces of data in order to determine whether they can be characterized as forming a group. These groups are termed as clusters.

In unsupervised classification, often known as 'cluster analysis' the machine is not told how the data are grouped. Its task is to arrive at some grouping of the data. In a very common of cluster analysis (K-means), the machine is told in advance how many clusters it should form. This is a potentially difficult and arbitrary decision to make.

## 4.4. Text Categorization Techniques

As described above the text categorization techniques are classified as supervised and unsupervised. In the following section each technique is described in details.

## 4.5. Supervised Text Categorization

The goal of supervised learning is to build a concise model of the distribution of class labels in terms of predictor features. The resulting classifier is then used to assign class labels to the

testing instances where the values of the predictor features are known, but the value of the class label is unknown. Supervised classification is the most common classification technique used in information systems. Many methods have been developed using artificial intelligence statistics. Logic and Perception based methods are developed by artificial intelligence where as Bayesian Networks and Instance based techniques are developed using statistics.

## a. Support Vector Method

Support vector method is a two way classification techniques. Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separation between a set of objects, having different class memberships. Decision planes are the classifiers either a line or a curve. A simple classifier may use liner decision planes rather than more complex structures. Classification tasks based on drawing separating lines to distinguish between objects of different class memberships are known as hyperplane classifiers.

Support Vector Machine (SVM) is primarily a classier method that performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables.

To construct an optimal hyperplane, SVM employees an iterative training algorithm, this is used to minimize an error function. According to the form of the error function, SVM models can be classified into distinct groups.

In the simplest SVM, training involves the minimization of the error function,

$$\frac{1}{2}w^T w + C\sum_{i=1}^{N}\xi_i$$

Subject to the constrains

$$y_i\left(w^T\phi(x_i)+b\right)\geq 1-\xi_i \text{ and } \xi_i \geq 0,\ i=1,\ldots,N$$

where C is the capacity constant, w is the vector of coefficients, b a constant and $\xi_i$ are parameters for handling non-separable data (inputs). The index i label the N training cases. Note that $y \in \pm 1$ is the class label and xi is the independent variables. The kernel $\phi$ is used to transform

data from the input (independent) to the feature space. It should be noted that the larger the C, the more the error is penalized. Thus, C should be chosen with care to avoid over fitting.

The success of SVM in text categorization lies in its automatic capacity tuning by minimizing||w||, extraction of a small number of support vectors from the training data that are relevant for the classification (Kwok, 1998). SVM in text categorization is a problem of very high dimension. Since the documents topics are not mutually exclusive, text categorization is usually analyzed as a serious of dichotomous classification problem, i.e whether the document belongs to a particular topic or not.

## b. K-nearest Neighbor Classification

The kNN classifier is based on the assumption that the classification of an instance is most similar to the classification of other instances that are nearby in the vector space. Compared to other text categorization methods such as Bayesian classifier, kNN does not rely on prior probabilities, and it is computationally efficient (Han et al., 1999). The main computation is the sorting of training documents in order to find the k nearest neighbors for the test document.

To classify a class-unknown document X, the k-Nearest Neighbor classifier algorithm ranks the document's neighbors among the training document vectors, and uses the class labels of the k most similar neighbors to predict the class of the new document. The classes of these neighbors are weighted using the similarity of each neighbor to X, where similarity is measured by Euclidean distance or the cosine value between two document vectors. The cosine similarity is defined as follows:

$$sim(X, D_j) = \frac{\sum_{t_i \in (X \cap D_j)} x_i \times d_{ij}}{\|X\|_2 \times \|D_j\|_2} \quad (3)$$

where X is the test document, represented as a vector; $D_j$ is the j[th] training document; $t_i$ is a word shared by X and $D_j$; $x_i$ is the weight of word $t_i$ in X; $d_{ij}$ is the weight of word $t_i$ in document $D_j$; $\|X\|_2 = \sqrt{x_1^2 + x_2^2 + x_3^2 + \dots}$ is the norm of X, and $\|D_j\|_2$ is the norm of $D_j$.

A cutoff threshold is needed to assign the new document to a known class.

k-nearest neighbor (k-NN) classification is an instance-based learning algorithm that has shown to be very effective in text classification. The success of this scheme is due to the availability of effective similarity measures such as cosine measure. However, the effectiveness of these similarity measures becomes worse as the number of words increases.

## c. Naïve Bayes

The Naive Bayes Classifier technique is based on the so-called Bayesian theorem and is particularly suited when the dimensionality of the inputs is high.

Let R = $\{r_1, r_2, r_3, \ldots. r_n\}$ denote the set of training open-ended responses, where each response is labeled with one of the cording in C = $\{c_1, c_2, c_3, \ldots. C_k\}$.

Given some new response, the aim is to estimate the probability of each code. Using Bayes rule, in general

$$p(c/r) = \frac{p(r/c)p(c)}{p(r)}$$

Since one is only interested in the relative order of the codes probabilities (given r) and by definition, p(r) is independent of C, one can focus on;

$$p(c/r) = p(r/c)p(c)$$

If denote the ordered sequence of unique words that compose the response r by

r=$\{w_1, w_2, w_3, \ldots..,w_p\}$

Then one can write,

$$p(r/c) = \prod_{i=1}^{p} p(w_i/w_1, w_2, w_3, \ldots\ldots, w_{i-1}, c)$$

However, using the naïve bayes assumption, we assume that the probability of each word in a response is independent of its context (Murphy K, 2006). More formally use the following approximation ("bag of words" model)

$$p(w_i/w_1, w_2, w_3, \ldots\ldots, w_{i-1}, c) = p(w_i/c)$$

Such that

$$p(r/c) = \prod_{i=1}^{p} p(w_i/c)$$

Thus to estimate $p(c/r)$ all need to estimate $p(w/c)$ and $p(c)$, for all words and for all codes. Use the following to estimate $p(c)$

$$p(c) = \frac{n(r,c)}{\sum_{c \in C} n(r,c)}$$

Where $n(r, c)$ is the number of training responses in the code c. The conditional probabilities of the words in c is estimated by

$$p(w_i/c) = \frac{n(c,w)}{\sum_{w \in W} n(c,w)}$$

Where $n(c, w)$ is estimated by

$$n(c,w) = \sum_{r \in R} n(r,w)$$

Where $n(r, w)$ is number of occurrences of the word w in the response r which is coded as c.

Then,

$$p(c/r) = p(r/c)p(c) = p(c) \prod_{i=1}^{p} p(w_i/c)$$

And classify r into the most possible code c using

$$\underset{c_j}{argmax}\ p(^{c_j}/_{r})$$

## d. Positive Naïve Bayes

The basic theme of the Naïve Bayes is to use joint probabilities of words and codes to estimate the probabilities of code given a response. The condition that the presence of each word in a response is conditionally independent of all other words in the response given its code is the, naïve assumption in this classification.

In positive Naïve Bayes we consider only two classes $\{0,1\}$ where 1 represent the positive class. Therefore the positive Naïve Bayes is problem of binary classification. As in the Naïve Bayes, consider a bag of words $w_1$, $w_2$, $w_3$,…….$w_n$, the unique word set retrieved from the responses. Then the positive Naïve Bayes classifier classifies a response r considering the n unique words as member of the class,

$$\underset{c \in \{0,1\}}{argmax}\ p(c) \prod_{i=1}^{n} p(w_i/c)$$

Given a word $w_i$ of the key word list in a two class classification problem, it can be from a document which was classified as positive (1) or negative (0).

Then P ($w_i$) satisfied the following equation;

$$p(w_i) = p(w_i/1) * p(1) + p(w_i/0) * p(0)$$

Hence $p(w_i/0)$ is calculated using the equation,

$$p(w_i/0) = \frac{p(w_i) - p(w_i/1) * p(1)}{1 - p(1)}$$

Since we consider only the positive examples the negative word probability, $p(0)$ is estimated by 1- $p(1)$.

Assuming that the set of unlabeled responses generated according to the underline generated model the word probability $p(w_i)$ is estimated by

$$p(w_i) = \frac{N(w_i, UR)}{N(UR)}$$

Where UR, is a set of unlabeled responses in the training set. N ($w_i$,UR), the number of unlabeled responses that contains the unique word $w_i$ and N(UR) is total number of unlabeled responses.

It is clear, if the word $w_i$ is not present in the open ended response considered then $p(w_i)$ becomes zero. Hence estimate of $p(w_i/0)$ is negative. We set the negative values to 0 and normalize the estimates such that the sum to 1. We smooth the estimates of negative word probabilities by dividing the number of unique words (Denis F. et al., 2002).

## 4.6. Unsupervised Text Categorization

### a. Hierarchical Clustering in text Categorization

A detailed description of the hierarchical clustering was presented in the chapter 3 section 3.5. In text categorization research, most of the studies have focused on flat classification where the predefined categories are considered for classification and there is no structure defining the relationships among them (Sun and Lim, 2001). Such categories are also known as flat categories. However, when the number of categories grows to a significantly large number, it will become much more difficult to cluster and classify the categories.

Hierarchical classification allows us to address a large classification problem using a divide-and-conquer approach. At the root level in the category hierarchy, a document can be first classified into one or more sub-categories using some flat classification method(s). The classification can be repeated on the document in each of the subcategories until the document reaches some leaf

categories or cannot be further classified into any sub-categories. A few hierarchical classification methods have been proposed recently. In most of the hierarchical classification methods, the categories are organized in treelike structures. On the whole, we can identify four distinct category structures for text classification. They are: Virtual category tree, Category tree, Virtual directed acyclic category graph, and Directed acyclic category graph.

## b. K-means Clustering in text Categorization

K-means algorithm is one of the most widely used central clustering techniques (Ghwanmeh S, 1998). A comprehensive explanation of building clusters using K– means given in the chapter 3 section 3.6.1.

The quality of the document list produced after classification depends on the number of clusters. Indeed, K-Means like methods require some a-priori decisions about the number of clusters. It is critical but not so easy to determine the number of clusters even if we have shown that it could be computed effectively according to the requirement. This is the main drawback in this clustering technique.

# Chapter 5: Hierarchical Clustering of Open-Ended Reponses

## 5.1 Chapter Objective:

This chapter describes the proposed solution for open-ended response categorization using hierarchical clustering. A natural language processing approach is proposed and strengthens the algorithm by introducing the document clustering techniques. Following sections discuss the main steps of the algorithm: Preprocessing, eliminating stops words, filtering the highest frequent words, Morphological Parsing, Vector and Vector representation and hierarchical Clustering.

## 5.2 Preprocessing

Preprocessing describes a type of processing on raw data before it input to the main processing procedure. Commonly used: data transformation, noise removal and normalization as the preprocessing techniques. In this algorithm the most required preprocessing task is cleaning the responses. The cleaning such as removing the punctuation marks, correcting the spelling mistakes and removing gibberish are to be completed at this stage. The punctuation marks do not carry any meaning in unstructured text like open-ended responses. Since most of the responses are short texts it is justified removing these punctuation marks prior to the next analysis. The algorithm considers the bag of words as the key feature of the algorithm then the unique word list is crated after applying the spelling correction. The words without any meaning, the gibberish are removed in preprocessing stage to complete the cleaning of open-ended responses.

### 5.2.1 Eliminating the Functional/Stop words

Function (grammatical) words are words which have little meaning but essential to maintain the grammatical relationships with other words. Function words also known as stop words include Prepositions, Pronouns, auxiliary verbs, conjunctions, grammatical articles or particles. For a given language the set of function words is closed and freely available. In text analysis these words are dropped in order to reduce the dimension of the feature vector. Since these function words carry less importantnce to meaning it is reasonable to remove all. But in this work the stop

words of type not, no don't, etc were not removed. The sense of these words affects the total meaning and the scale of the responses.

### 5.2.2 Filtering the highest frequent words

The feature vector of the proposed algorithm is based on the unique key word list. The unique word list is constructed by considering the frequency of the content words (non-function) and then the highest frequency words are filtered using a suitable cut-off frequency. The cut of frequency is decided by applying the zip's law (Manning & Schutze 1999) for the entire word list of the selected responses. The reverse engineering process also verifies the cutoff. This highest frequent word list helps to reduce the search space of the responses and it will be the vector-space representation.

### 5.2.3 Morphological Parsing

The filtered unique word list contains many morphological forms of the stem word. To normalize the words in to a single form such as stem the morphological parsing is required. Hence all the responses (key words) undergo morphological passing in order to remove the inflectional and derivational morphemes of the non-functional words. The morphemes like Plurals, Continuous, past, etc are removed in this process.

## 5.3 Vector and Vector representation

The clustering algorithm requires a vector based representation of the responses to classify the responses. The Vector is constructed using the highest frequent word list. There are many ways to identify the features for the vector. It may be any kind of distance between the selected key word list and the responses. The most popular and key word based measurement for this type of study is tf-idf weight. Where tf denotes the term frequency for the responses which is simply the number of times a given term appears in that response. This value is normalized to avoid the bias to long responses to give the exact importance of the word. And it is calculated using the following equation

$$\mathrm{tf_{i,j}} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

Where $n_{i,j}$ is no of times the term $t_i$ appears in the response $d_j$ and the denominator is the sum of all the words in the response $d_j$.

The inverse document frequency (idf) is a measure of the general importance of the term. Idf is obtained by dividing the number of all responses by the number of responses containing the term. Then the logarithm of the quotient calculated as

$$idf_i = log \frac{|D|}{|\{j: t_i \in d_j\}|}$$

Where |D|total number of responses considerd and $|\{j: t\_i \in d\_j\}|$ is no of responses where the term $t_i$ is appears. The division-by-zero occurs when the term $t_i$ is not present in the responses. To avoid this one can change the denomenator to $1 + |\{j: t_i \in d_j\}|$

Then the

$$(\text{tf-idf})_{i,j} = \text{tf}_{i,j} \times \text{idf}_i$$

Other alternative is binary representation. The binary vector for each and every response is constructed by considering the presence and absence of highest frequent word. As for the tf-idf the size of the binary vector is the length of the highest frequent word list.

## 5.4 Application Hierarchical Clustering for Open-ended Responses

The input to the hierarchical clustering algorithm is vectors constructed in the above step. As described in chapter 3 in Hierarchical clustering the similarities between vectors are calculated using cosine similarity and the linkage function use in the algorithm is "Average". Cosine similarity measures a similarity between two vectors by considering the angle between the vectors. If two vectors are completely same then the angle between the vectors is equal to 0 and hence the cosine similarity value is 1 or-1(completely opposite). Since the (tf-idf) value is positive the case of -1 is does not occur. If two vectors are orthogonal then the similarity value is 0. In between these values indicates the similarity and the dissimilarity of the vectors.

As mentioned in chapter 3 Hierarchical clustering algorithms are either top-down or bottom-up. Bottom-up algorithms treat each response as a singleton cluster at the outset and then

successively merge (or *agglomerate*) pairs of clusters until all clusters have been merged into a single cluster that contains all responses. Top-down clustering requires a method for splitting a cluster. It proceeds by splitting clusters recursively until individual responses are reached. A dendrogram is constructed to illustrate the clusters produced by the hierarchical clustering. Hierarchical clustering does not require a pre specified number of clusters. However, in this application the author wants a partition of disjoint clusters just as in flat clustering. In this case, the hierarchy needs to be cut at some point. A number of criteria can be used to determine the cutting point of the dendrogram

i. Cut at arbitrary level of similarity
ii. Cut the dendrogram where the gap between two successive combination similarities is the largest. Such large gaps indicate ``natural'' clusterings. Adding one more cluster decreases the quality of the clustering significantly, so cutting before this steep decrease occurs is desirable.
iii. As in flat clustering, pre-specify the number of clusters k, and select the cutting point that produces k clusters.

In this algorithm author uses the criteria iii since the number of codes for a given set of responses is pre defined and can be used the number clusters required.

The cluster evaluation is done by the “cophenetic” correlation coefficient. Cophenetic Correlation Coefficient is the correlation between distances of each vector with the minimum distance calculated to merge the vectors in hierarchical clustering method. As in the case of simple correlation, if the Cophenetic Correlation Coefficient is 100% indicates the best fit in clustering (Johnson and Wichern, 1996).

## 5.5 The Graphical Overview of the system:

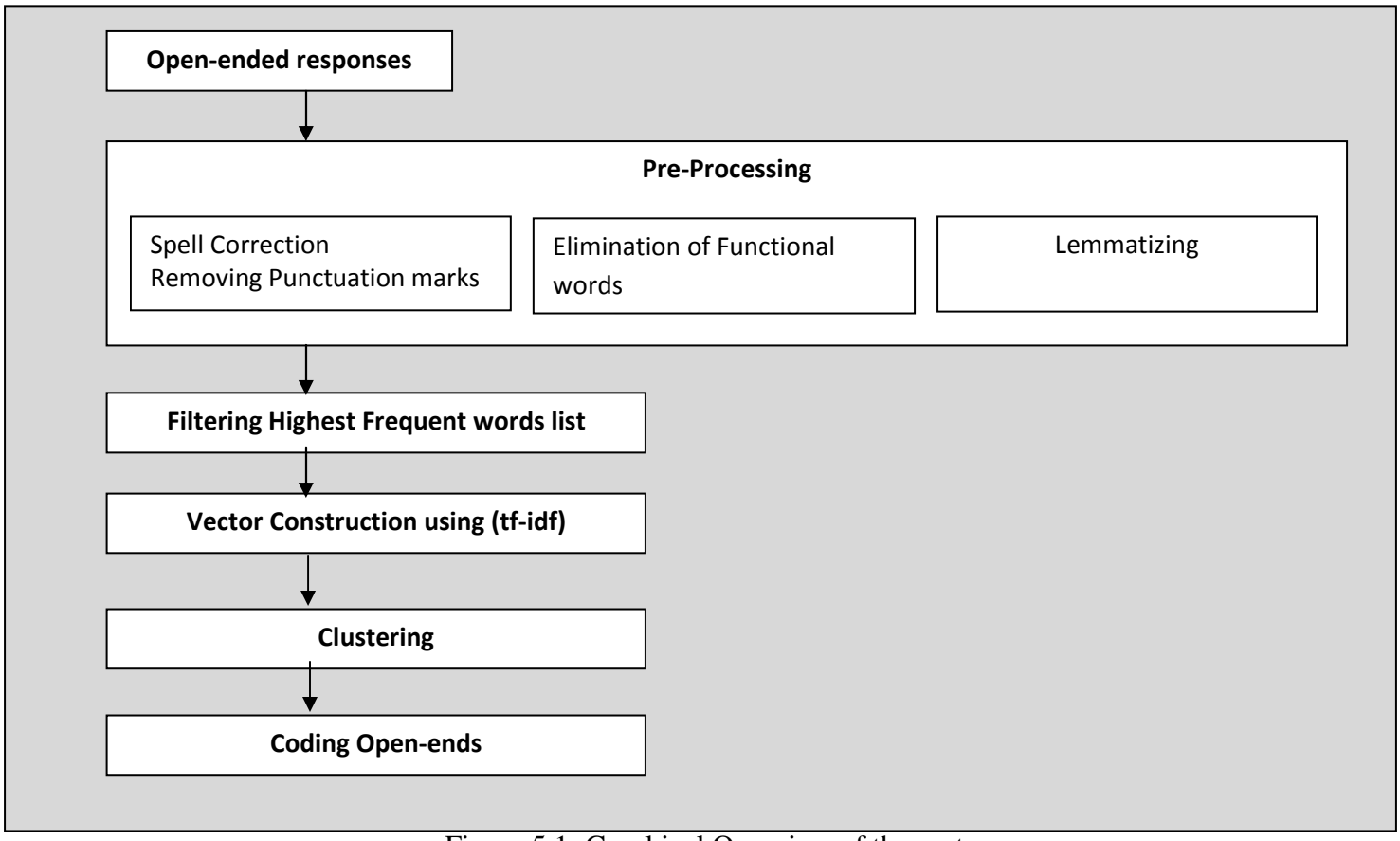



Figure 5.1: Graphical Overview of the system

# Chapter 6: Categorizing Open-Ended Responses using a Naïve Bayes Model

## 6.1 Chapter Objective:

This chapter explains the algorithm of Naïve Bayes classification on open-ended response categorization. The attempt of classical Naïve Bayes described in the first part of the chapter and Naïve bayes application in positive and unlabeled examples is discussed in the latter part of the chapter. The reason for testing the Naïve Bayes for positive and unlabeled examples is explained in detail in the following chapter.

## 6.2 Why Naïve Bayes for open-ended response categorization:

As mentioned in chapter 2 there are two ways of categorizing open-ends. In most of the instances all the responses are coded by the researcher one by one, visually going through each response. This is time consuming and very pain full exercise for the researcher. In some cases part of the responses are coded as earlier but predefined set of code is used to code them. In this case the researcher categorizes the sample of responses using the predefined codes prior to code the whole responses. The set of responses which are coded by the researcher can be considered as a training data set for naïve bayes classification and then we can use naïve bayes learning to code the rest of the responses. This means we have a set of training and test data for the classification. Hence a Naïve Bayes classification is applicable in open-ended response categorization as well.

## 6.3 Preprocessing:

As described in the previous chapter the preprocessing is an essential task in text analysis. Hence preprocessing is important as the first step of naïve bayes classification. The tasks of, eliminating stops words, filtering the highest frequent words, morphological parsing, were carried out to select the key word list. As in the hierarchical method described in the chapter 5 the highest frequent word list contain unique words with frequency greater than or equal to 3. The morphological variations of highest frequent words are removed and the key word list includes stem of such words.

## 6.4 Training Data for Naïve Bayes

Naïve Bayes is a supervised training classification algorithm. Hence this requires training and test data for the classification. In this experiment the requirement is to test the Naïve Bayes in open-ended response categorization with minimum number of training data set. As the manually coded responses are limited the training data set consists of less number of coded responses. Therefore we select 38% of the responses as the training data set where as the rest of the responses are reserved to test the classification accuracy.

## 6.5 Calculating the word probabilities

The posterior probability of a code c given a response r is the main estimation of Naïve Bayes classification to code the response as c. As in chapter 4 section 4.5 the probability is estimated by the product of likelihood of code c and the prior of response r given the code c. Using the Naïve Bayes assumption the prior probabilities for the response r are calculated using the key word list contain in the response. Therefore for each keyword wi, the conditional probability of $\mathrm{p}(\mathrm{w_i}/\mathrm{c})$ is estimated using

$$p(w_i/c) = \frac{n(c,w)}{\sum_{w \in W} n(c,w)}$$

Where $\mathrm{n}(\mathrm{c},\mathrm{w})$ is estimated by

$$n(c,w) = \sum_{r \in R} n(r,w)$$

Where $\mathrm{n}(\mathrm{r},\mathrm{w})$ is number of occurrences of the word w in the response r which is coded as c.

The likelihood $p(c)$ is the probability of code c and it is the relative frequency of number responses coded as c to total number of responses coded.

With the naïve assumption the posterior probability is calculated as

$$p(r/c) = \prod_{i=1}^{p} p(w_i/c)$$

## 6.6 Coding the test responses using Naïve Bayes:

The entire preprocessing tasks were carried out for all the test responses prior to classifying them. The posterior probability for each response is calculated considering occurrence of key words in the response. The code relevant to maximum posterior probability of the given response is assigned to the response as the probable code.

## 6.7 Positive Naïve Bayes for open-ended responses:

One of the drawbacks of applying the classical Naïve Bayes in open-ended categorization is the lack of training data. Since the high cost of training data, the categorization is to be carried out with the minimum size of training data. The solution for such situation is the application of Positive naïve bayes or incremental learning. Positive Naïve Bayes is binary classification algorithm with less number of training data set. In this study we accomplish the positive naïve bayes considering the category of most significant as the positive case and the training data set consists of 30% of the total responses available. The prior and posterior probabilities are calculated according to the chapter 4 section 4.5. After coding the responses using the positive code set the second run of the positive Naïve Bayes is carried out to code the balance (negative) responses. In this case the next most significant code is set as the positive example. By continuing in this nature all the responses were coded selecting the positive code at each run.

# Chapter 7: Experiment & Evaluation - Hierarchical Clustering

## 7.1 Chapter overview

This chapter explains the application of the algorithm discussed in the previous chapter for several open-ended responses gathered in various statistical researches. This includes the application in several domains such as market research, social research and for opinion extraction. Selection of responses for the study, Evaluation method and application of the techniques described in the thesis to a real world problem was described comprehensively. Finally the method of evaluation of the categories obtained is discussed in detail.

## 7.2 Selection of responses for categorization

Algorithm is tested for different types of responses gathered in various statistical surveys. The responses considered in this study were categorized in to several groups by considering the domain of the open-ended question asked in the survey. Mainly the responses classified as

I. Opinions on customer satisfaction in a certain product
   This category consists of responses about products like "Mobile Phones', "Hamburgers" etc. The length of the responses in this group is shorter than the other and contains maximum 10 words.

II. Responses on social science claim such as environment, work place satisfaction. These responses include long sentences. In some cases, responses contain 3 to 4 sentences.

## 7.3 Evaluation method for Text Categorization

Evaluation of the codes obtained by the proposed algorithm was carried out by comparing the manually corded responses. Therefore the study was conducted with set responses those were coded before by the domain expert.

Text Categorization can be evaluated using Precision, Recall and F-measure. These standard measures have significantly higher correlation with human judgments (Manning et al., 2009). These are first defined for the simple case where a text categorization system returns the categories.

Precision (P) is the fraction of retrieved documents that are relevant

$$Precision = \frac{\#(relevent\ items\ retrived)}{\#(retrieved\ items)} = P(relevant|retrived)$$

Recall (R) is the fraction of relevant documents that are retrieved

$$Recall = \frac{\#(relevent\ items\ retrived)}{\#(relevent\ items)} = P(retrived|relevant)$$

These notions can be made clear by examining the following contingency table;

**Table 7.1 Precision and Recall Contingency table**

| | Relevant | Non-relevant |
|---|---|---|
| Retrieved | true positives (tp) | false positives (fp) |
| Not retrieved | false negatives (fn) | true negatives (tn) |

Then;

$$P = \frac{tp}{(tp + fp)}$$

$$R = \frac{tp}{(tp + fn)}$$

The measures of precision and recall concentrate the evaluation on the return of true positives, asking what percentage of the relevant documents has been found and how many false positives have also been returned.

A single measure that trades off precision versus recall is the F measure, which is the weighted harmonic mean of precision and recall. F score is a measure of a test's accuracy. There are different weights that can be calculated for F measure. The balance F measure equally weights precision and recall and it is commonly written as $F_1$

$$F_1 = \frac{2PR}{(P+R)}$$

Then, the $F_1$ score can be interpreted as a weighted average of the precision and recall, where an $F_1$ score reaches its best value at 1 and worst score at 0.

## 7.4 An Application in product base responses

### 7.4.1 The study and its Responses

The proposed algorithms was tested on the responses collected in a survey of about environmental pollution poses the open-ended question "How do you think the roles of The Energy Savings Trust and The Carbon Trust differ?". Out of 158 respondents of the above study, 155 have responded to this question. Nine of the responses given are gibberish and are hence removed from the input list. There are ten manual codes for the responses coded by the researcher and the human classification is given in Table 7.2 below.

**Table 7.2: Manual Codes**

| Category Id | Response | Frequency |
|---|---|---|
| 1 | Energy Saving Trust has a wider brief/more general remit - Carbon Trust is more specific | 8 |
| 2 | Energy Saving Trust gives advice on saving energy generally - it considers all types of energy | 35 |
| 3 | Carbon Trust relates only to carbon | |
| 4 | Carbon Trust tries to reduce our carbon footprint/tries to reduce our level of carbon emissions | 3 |
| 5 | Carbon Trust trades carbon/offsets our carbon footprint | 1 |
| 6 | Mention of concern about impact on the environment | 6 |
| 7 | Mention of concern about saving money | 6 |
| 8 | They are the same/don't differ/very little difference | 5 |
| 9 | Other | 7 |
| 10 | Don't Know | 77 |

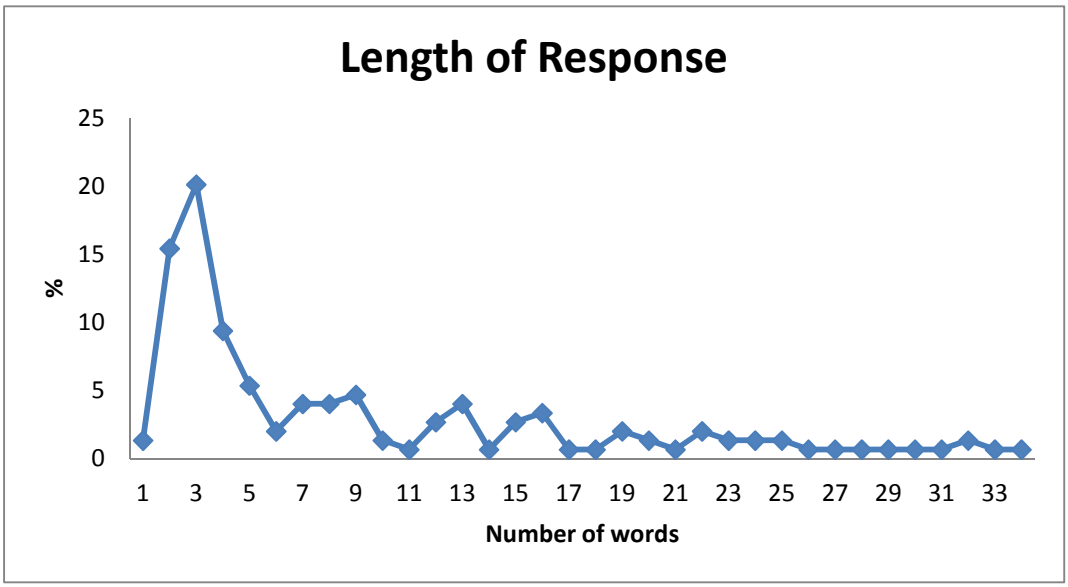
The length of the responses is varying from 1 to 49 in words. The minimum length is one word and there is one response with maximum length of 49 words. About 35% of the responses are minimum length of 2 words. Length of the responses distributed as in the table 7.3.

**Table 7.3: Length of Responses**

| Length of the Responses (in words) | Frequency | % |
|---|---|---|
| 3 | 30 | 20.134 |
| 2 | 23 | 15.436 |
| 4 | 14 | 9.396 |
| 5 | 8 | 5.3691 |
| 9 | 7 | 4.698 |
| 7 | 6 | 4.0268 |
| 8 | 6 | 4.0268 |
| 13 | 6 | 4.0268 |
| 18 | 5 | 3.3557 |
| 12 | 4 | 2.6846 |
| 15 | 4 | 2.6846 |
| 6 | 3 | 2.0134 |
| 21 | 3 | 2.0134 |
| 24 | 3 | 2.0134 |

The graphical representation of the length distribution as figure 7.1.

Figure 7.1: Graph of Response Length




In this experiment there are 10 manual codes defined by the researcher and all the responses were coded by these 10 codes.

**Table 7.4: Set of manually coded responses**

| Response | Manual Code |
|---|---|
| EST wider brief than CT | **1** |
| energy savings trust has a wider remit. | 1 |
| Energy savings less specific than carbon Trust | 1 |
| One is to save energy the other to reduce greenhouse gases | 2 |
| One saves energy, the other looks after carbon | 2 |
| first one is to reduce amount of energy used second is to reduce carbon footprint | 2 |
| very little | 8 |
| They Are Both The Same | 8 |
| Very little | 8 |
| Sorry no idea | 10 |
| dont know | 10 |
| I dont know | 10 |
| have no idea | 10 |

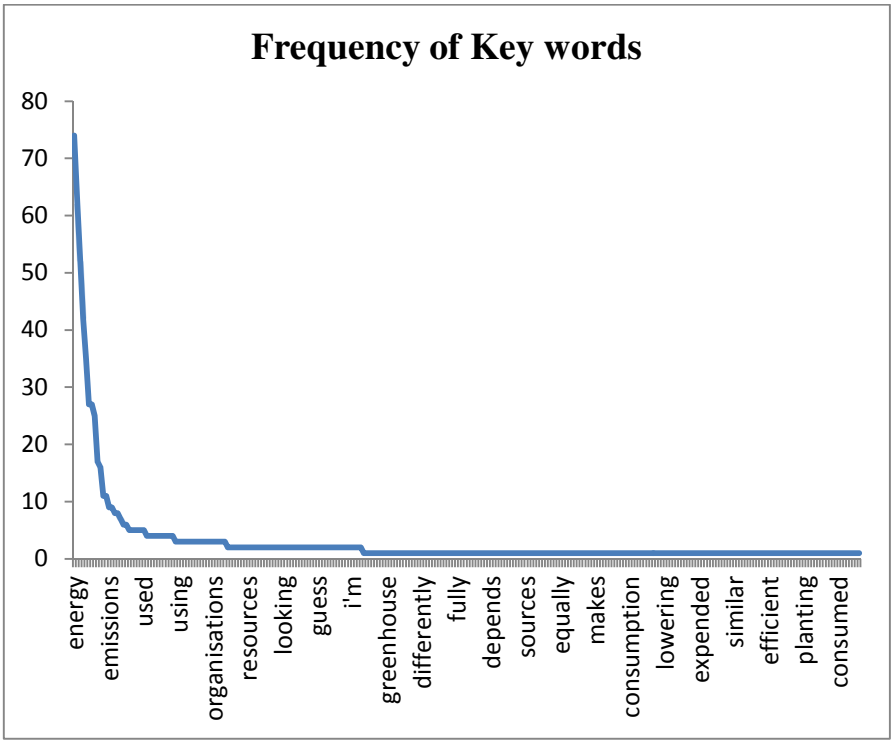
### 7.4.2 Removing functional words and retrieving the key word list

Algorithm has been applied on the response set mentioned in 7.4.1, after completing the preprocessing task as described in chapter 5. Initially there were 272 unique (key) words retrieved by eliminating the functional words. The standard functional word list available in the web is used to retrieve the key word list. But considering nature of the current responses the stop words pertaining to negation such as “not”, “no” and “don’t”, were not removed since these words affect the total meaning and the polarity of the responses. Following is the most frequent key words distribution (words of frequency greater than or equal to 5).

**Table 7.5: Key word distribution**

| Key Word | Frequency | % |
|---|---|---|
| energy | 74 | 8.79 |
| carbon | 62 | 7.36 |
| not | 52 | 6.18 |
| trust | 42 | 4.99 |
| know | 35 | 4.16 |
| idea | 27 | 3.21 |
| no | 27 | 3.21 |
| saving | 25 | 2.97 |
| savings | 17 | 2.02 |
| reducing | 16 | 1.90 |
| save | 11 | 1.31 |
| use | 11 | 1.31 |
| emissions | 9 | 1.07 |
| footprint | 9 | 1.07 |
| environment | 8 | 0.95 |
| reduce | 8 | 0.95 |
| looks | 7 | 0.83 |
| usage | 6 | 0.71 |
| est | 6 | 0.71 |
| money | 5 | 0.59 |
| impact | 5 | 0.59 |
| focused | 5 | 0.59 |
| sure | 5 | 0.59 |
| heard | 5 | 0.59 |
| used | 5 | 0.59 |

**Figure 7.2: Graph of Key word Frequency**




The unique key word list contains words with various morphological derivatives of some words. The plural derivatives like “savings” and verb forms like “focused” are some of morphological variations that can be observed in the key word list. To generalize these words to a unique word the process of lemmatizing is required. It is assume that, the information of having these variations will be lost by removing the morphological derivations. Therefore the experiment is carried out with all forms of the words. According to the figure 7.1 the frequency of the key words drops steeply from 74 to 3.

### 7.4.3 Selecting unique word list for the vector

As described in chapter 6 a vector of key words is formed by selecting a set of words in unique words list retrieved. This set of words was selected based on the frequency of the words. Since there are 272 key words in the list the criteria of selecting the set is the highest frequent words. These highest frequent values vary from 1 to 74. The threshold of selecting the words was decided by applying the reverse engineering technique. A set of several tests is carried out by varying the threshold value from 2 to 5. Then the evaluation was carried out using the manual categorization of the responses. The F value is calculated for each test. Following are F measures, and it's clearly indicated as the highest frequency of the key word list increases the F value decreases gradually.

**Table 7.6: F values**

| | # key words | Precision | Recall | F |
|---|---|---|---|---|
| Key words of frequency 5 or more | 19 | 0.8125 | 0.382352941 | 0.5199999 |
| Key words of frequency 4 or more | 25 | 0.611111111 | 0.647058824 | 0.628571429 |
| Key words of frequency 3 or more | 35 | 0.609756098 | 0.714285714 | 0.657894737 |
| Key words of frequency 2 or more | 53 | 0.590909091 | 0.742857143 | 0.658227848 |

The table 7.6 reveals that when the key word frequency increases the precision increases proportionally. Whereas recall drops as the key word frequency is increased. As there is no much significant difference of F value in both frequencies 3 and 2, it is decided as the size of the key word list to be the words with minimum frequency 3. This count further justified by the Zip's as the figure 7.1 shows the deviating the curve at frequency 3.

### 7.4.4 Generating the feature vector

As per the section5 6.3 in chapter 5 the feature vector was generated using tf-idf values. The vector consists of 35 columns of key words and constructed for all the responses. The responses considered for vector generation were cleaned and no functional words. As idf calculations depend on the length of the responses, all functional words were removed.

**Table 7.7: An illustration of generated vectors**

| | **energy** | **carbon** | **not** | **trust** | **know** | **idea** | **no** | **saving** | **savings** | **reducing** | **save** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Response 1** | 0 | 0 | 0.523 | 0 | 0.721 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Response 2** | 0.178 | 0.206 | 0.087 | 0.148 | 0 | 0 | 0 | 0 | 0.180 | 0 | 0 |
| **Response 3** | 0 | 0 | 0.523 | 0 | 0.721 | 0 | 0 | 0 | 0 | 0 | 0 |

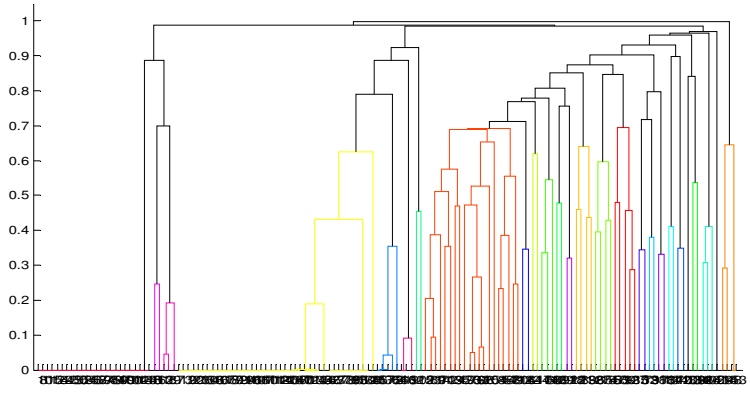
| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Response 4** | 0.152 | 0.177 | 0 | 0 | 0 | 0 | 0 | 0.286 | 0 | 0.385 | 0 |
| **Response 5** | 0.178 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Response 6** | 0.213 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.520 |
| **Response 7** | 0.266 | 0.309 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Response 8** | 0 | 0 | 0 | 0 | 0 | 0.851 | 0.851 | 0 | 0 | 0 | 0 |
| **Response 9** | 0 | 0 | 0.523 | 0 | 0.721 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Response 10** | 0 | 0 | 0 | 0 | 0 | 0.851 | 0.851 | 0 | 0 | 0 | 0 |

Table 7.7 shows the sample of tf-idf value calculated according to the chapter 6 section 6.3. The tf-idf value was calculated for each response to each key word selected. The key word that does not include in the open ended response gives the tf-idf value to 0.

### 7.4.5 Generating Clusters

Feature vector generated in above step is clustered by hierarchical clustering method described in chapter 6. As per the section 5.4 the similarity measure is cosine similarity and average linkage selected as the linkage function. Selection of linkage function based on the similar reverse engineering process explained in section 7.4.3. The vectors containing all zeros were removed before doing the clustering since the division by zero error is encountered when calculating the similarity of such vectors. Then the dendrogram pertaining to the hierarchical clusters were obtained. The strong cophenetic correlation coefficient (0.96428) shows the validity of the hierarchical clustering and the accuracy of linkage function.

Figure 7.2: Dendrogram for the clusters




Then the clusters were obtained by cutting the dendrogram at constant height. Since there are 10

manual codes defined in the experiment it is not unreasonable to set the cutoff of the dendrogram to be 10. By setting the number of clusters to 10, responses were clustered to 10 clusters.

According to the table 7.2, out of 148 responses, about 24% of them are coded as "2" and 52% were coded as "10". Therefore these categories are significant and important to the study of opinion of "Energy Savings Trust and The Carbon Trust difference". Thus the author analyzed these two categories extensively and presented the findings and evaluation in following sections. Followings are the set of responses with retrieved by the proposed algorithm pertaining to the manually code 2.

**Table 7.8: Reponses extracted as code 2**

| |
|---|
| Not much. Energy Savings considers all types of energy. The Carbon Trust is about carbon output. |
| one is concerned with energy saving the other with reducing overall carbon emissions |
| One is to save energy the other to reduce greenhouse gases |
| One saves energy, the other looks after carbon |
| first one is to reduce amount of energy used second is to reduce carbon footprint |
| the energy saving trust concentrates on ways of saving energy and the carbon trust looks at doing things differently which will help reduce the carbon imprint |
| Energy Saving Trust is to do with minimizing the use of energy and ensuring people use it more efficiently to save resources. The Carbon Trust is about minimizing the impact of those carbon based products that are used |
| Energy Saving Trust is about saving energy and being more economic with the way we use it. Carbon Trust is about our footprints |
| One aims to save energy, the other trades carbon |
| One is for the energy saving which is essential for human begin another is for the saving of world form depletion of carbon dioxide |

The highest key words in this list of responses are (more than 5 frequency) "energy, carbon, trust, saving, reducing, save, use, reduce, environment, looks, emissions, usage". The comparison of these key words with the total key words list is more important to test the suitability of the cluster obtained by the algorithm.

**Table 7.9: Comparison of keywords**

| **From All Responses** | | **From Cluster 2 Responses** | |
|---|---|---|---|
| **Word** | **Frequency** | **Word** | **Frequency** |
| no | 79 | - | - |
| energy | 74 | energy | 65 |
| carbon | 62 | carbon | 57 |
| save | 54 | save | 10 |
| trust | 42 | trust | 39 |

| | | | |
|---|---|---|---|
| know | 35 | - | - |
| idea | 27 | - | - |
| reduce | 24 | reduce | 8 |
| use | 22 | use | 10 |
| emission | 9 | - | - |
| footprint | 9 | - | - |
| environment | 8 | environment | 7 |
| looks | 7 | looks | 7 |
| est | 6 | - | - |
| - | - | saving | 24 |
| - | - | reducing | 16 |
| - | - | emissions | 7 |
| - | - | usage | 6 |

The table 7.8 shows the words meaning to the negation is not included in the responses clustered as code 2 by the algorithm. This reveals the cluster is completely disjoint with the cluster of code 10 whose responses having such negation words. Also this keywords comparison shows the requirement of getting the stem of words by applying the lemmatizing techniques.

To further evaluate this cluster, precisian and recall values were calculated.

**Table 7.10: Precision and Recall**

| | Relevant to code 2 | Not Relevant to code 2 | Total |
|---|---|---|---|
| Retrieved as Cluster 2 | 26 | 18 | 44 |
| Not Retrieved as Cluster 2 | 9 | 92 | 101 |
| Total | 35 | 110 | |
| Precision | 0.590909091 | | |
| Recall | 0.742857143 | | |
| F | 0.658227848 | | |

The high F measure shows an accuracy of the retrieving cluster 2 and the total clustering process. Similar analysis had been carried for the next most important cluster relevant to the manual code 10. This cluster is totally with the idea of negation and unawareness of energy savings trust and carbon trust. The responses of code 10 were clustered by the algorithm mainly into 3 clusters with “don’t know”, “no idea” and “Haven't heard” as the main key words in each cluster respectively. Out of 72 responses for manually coded as 10 , 43 retrieved as cluster 10 with key words “don’t know”, 24 retrieved as cluster 5 with key words “no idea” and 4 retrieved as cluster 6 with key words “Haven't heard”. The respective F measures are 0.717, 0.5 and 0.103.

Semantically all these three types of key words are same and the responses are closely related to the negation. So it is reasonable to consider the clusters with number 10, 5 and 6 obtained by the algorithm are similar. Considering these clusters together as a single cluster the F value is calculated. It shows a high significant value of 0.953.

### 7.4.6 Appling the stemming Algorithm

The key word list considered in the table 7.5 contains some words with its morphological variations as well. A test is carried out by removing the morphological forms of the set of key words that has the frequency greater than 10. The "ing" form in "saving" and "reducing" is removed and consider only the root form "save" and "reduce". The plural morpheme in "emissions" and "savings" are dropped. The stem "use" is considered instead of "usage" and "used". Then the algorithm is applied for stemmed responses. The F value of the code cluster has been increased significantly by applying the stemming algorithm.

**Table 7.11: F Measure after applying stemming algorithm**

| | Relevant to code 2 | Not Relevant to code 2 | Total |
|---|---|---|---|
| Retrieved as Cluster 5 | 31 | 21 | 52 |
| Not Retrieved as Cluster 5 | 4 | 89 | 93 |
| Total | 35 | 110 | 145 |
| Precision | 0.596153846 | | |
| Recall | 0.885714286 | | |
| F | 0.712643678 | | |

The algorithm with the process of lemmatizing shows promising results than the algorithm without the same.

### 7.4.7 Conclusion

This chapter explained the application of hierarchical clustering in open ended responses. Accuracy of obtaining clusters was between 65% to 95% without applying the stemming algorithm. A significant improvement of accuracy shows the importance of stemming algorithm in clustering open ended responses.

# Chapter 8: Experiment & Evaluation – Naïve Bayes Classification

## 8.1 Chapter overview

This chapter explains the application of Naïve Bayes classification algorithm on open-ended responses. The algorithm tested on several types of open-ended responses collected in various statistical surveys. The evaluation of the categories obtained is discussed with relevant measures.

## 8.2 Responses considered for categorization

Algorithm is tested for the responses collected by the survey of environmental pollution poses the open-ended question "How do you think the roles of The Energy Savings Trust and The Carbon Trust differ?". After cleaning the responses only 149 were selected for the algorithm. As in the case of open-ended response categorization the objective is to train the algorithm with minimum number of training set. As in the case of chapter 7, the decision of selecting the minimum number of training set was the cost of obtaining such label data set. Therefore 38% of the responses were considered for the training data set. The code distribution of the training data set is given in the table 8.1.

**Table 8.1: The Training Data set**

| Code | # Responses | % of Training set | % of Total sample |
|---|---|---|---|
| C1 | 3 | 5 | 38 |
| C2 | 15 | 26 | 43 |
| C4 | 2 | 4 | 2 |
| C5 | 1 | 2 | 0.7 |
| C6 | 2 | 4 | 4 |
| C7 | 3 | 5 | 4 |
| C8 | 3 | 5 | 3 |
| C9 | 4 | 7 | 5 |
| C10 | 24 | 42 | 52 |

The Table 8.1 described the contribution of each code to the training dataset. As an example 3 responses of code 1 selected for the training set. This is 38% of the total sample space and 5%

weight to the training set. C2 and C10 are the most significant clusters of the study and are given a high weight to represent the training set.

## 8.3 Calculating the Prior probabilities for Naïve Bayes classification

All the functional (stop) words contained in the selected training set were removed before calculating the probabilities. As in the hierarchical clustering approach in chapter 7 the negation words line "not" and "no" were excluded from the stop words list. Hence these words remained in the responses. The key word list is retrieved considering the frequency of non-functional words. The cut off frequency for the selection is set to 3 as the experiment in chapter 7 shows it is the optimum frequency of the words to be selected for the key word list. There were 35 key words selected by this criterion. The prior probabilities for the key word given a code are calculated as per the equation derived in the chapter 6 section 6.5. The likelihoods for the codes were the relative frequency of each in the training data set.

## 8.4 Testing the algorithm

The algorithm was tested with 92 responses by calculating the posterior probability for code given a response. The code given by maximum posterior probability is assigned to the response as its code. The algorithm returns only 3 codes out of 9 codes trained. The most significant codes are analyzed further with precision and recall measures.

**Table 8.2: Precision and Recall of code number 2**

| | Relevant to code 2 | Not Relevant to code 2 | Total |
|---|---|---|---|
| Retrieved as Cluster 2 | 8 | 8 | 16 |
| Not Retrieved as Cluster 2 | 12 | 64 | 76 |
| Total | 20 | 72 | 92 |
| Precision | 0.5 | | |
| Recall | 0.4 | | |
| F | 0.222222 | | |

The code shown in this cluster does not show good accuracy as the low value of the precisian and recall.

**Table 8.3: Precision and Recall of code number 10**

| | Relevant to code 10 | Not Relevant to code 10 | Total |
|---|---|---|---|
| Retrieved as Cluster 10 | 47 | 2 | 49 |
| Not Retrieved as Cluster 10 | 8 | 35 | 43 |
| Total | 55 | 37 | 92 |
| Precision | 0.959184 | | |
| Recall | 0.854545 | | |
| F | 0.451923 | | |

This code shows sufficient Precision and Recall values, but the F value is below the acceptable range.

The reason for getting the poor accuracy of the Naïve Bayes in open-ended responses categorization may be the size of the training set. The small training data set effect the learning model in this experiment for poor accuracy. As mentioned earlier the solution is incremental learning or the positive Naïve Bayes. In the following sections describe the attempt of positive Naïve Bayes with small training data set.

## 8.5 Positive Naïve Bayes experiment of open-ended responses.

The positive Naïve Bayes is binary classification small training data. In this study we consider the most significant codes of the responses explained in section 8.2. The code number 2 and 10 are the most frequent classification in this example. Hence we considered these codes for testing the positive Naïve Bayes. We begin experiment by taking code number 2 as the positive case and the others as the negative example. 13% of the responses which are coded as code number 2 were selected as the training set and rest of the responses are the unlabeled example of code 2.

**Table 8.4: Responses selected for Positive Naïve Bayes**

| Response for Positive Class |
|---|
| Not much. Energy Savings considers all types of energy. The Carbon Trust is about carbon output. |
| one is concerned with energy saving the other with reducing overall carbon emissions |
| One is to save energy the other to reduce greenhouse gases |
| One saves energy, the other looks after carbon |
| first one is to reduce amount of energy used second is to reduce carbon footprint |
| the energy saving trust concentrates on ways of saving energy and the carbon trust looks at doing things differently which will help reduce the carbon imprint |
| Energy Saving Trust is to do with minimizing the use of energy and ensuring people use it more efficiently to save resources. The Carbon Trust is about minimizing the impact of those carbon based products that are used |
| Energy Saving Trust is about saving energy and being more economic with the way we use it. Carbon Trust is about our footprints |

| |
|---|
| One aims to save energy, the other trades carbon |
| One is for the energy saving which is essential for human begin another is for the saving of world form depletion of carbon dioxide |
| The former is looking after usage of energy with the effect being savings to a business as well as to the environment whilst the latter looks to provide details of the impact on the environment of energy usage and so encourage energy efficiency |
| the former relates to the saving of energy and the latter's emphasis is on reduction of carbon emissions |
| Energy Saving Trust concerned with energy in general, Carbon Trust concerned with reducing carbon emissions e.g. carbon trust would favor nuclear generation as it is a low-carbon option but EST would argue that building new nuclear plants could be avoided by lowering demand for energy over all. |
| The EST are concerned with all aspects of energy saving whereas the CT major on energy savings of carbon based issues |
| the energy savings trust is interested in saving energy while the carbon trust concentrates on the amount of carbon in the environment. |

All the responses are long sentences with main key words like “energy”, “carbon”, “trust” ect. Stop words contained in the above responses are removed and key word list is selected as per same criteria used in classical Naïve Bayes. The root form of highest frequent words in the key word list obtained after removing the morphological variation such as plurals, past tense and “ing” forms.

The prior probabilities of the positive and negative classes were calculated according to the chapter 4 section 4.5.d. Then the classification was carried out for the test data set of 129 responses. The classification of code number 2 as below.

**Table 8.5: Precision and Recall for code 2-Positive Naïve Bayes**

| | Relevant to code 2 | Not Relevant to code 2 | Total |
|---|---|---|---|
| Retrieved as Cluster 2 | 10 | 33 | 43 |
| Not Retrieved as Cluster 2 | 5 | 81 | 86 |
| Total | 15 | 114 | 129 |

| | |
|---|---|
| Precision | 0.232558 |
| Recall | 0.666667 |
| F | 0.344828 |

The recall has been improved in positive Naïve Bayes classification over the classical naïve bayes. The F value is still below the standard as the Precision is affected.

Classification of code number 10 also carried out in similar ways as in the code 2. But the responses classified as code 2 are removed the sample before analyzing the code 10. Then the total sample consists of 114 responses. The training set for code 10 selected with 15 responses as the percentage of 10%. The following are the training response set.

**Table 8.6: Training Responses of code number 10**

| Response for Positive Class |
|---|
| do not know |
| No idea |
| no idea |
| Unaware of the difference |
| I Haven't a clue |
| Unsure |
| not heard of either of them |
| I do not have enough information to give an opinion |
| Haven't heard of either |
| do not know |
| not much |
| do not know |
| no idea |
| do not know |
| I have never heard of them. |

Most of the responses are short sentences with meaning of negation. The key word list selected in the same done in earlier cases and the negation term such as “Haven’t” are normalized to a general form of “Have not”. The testing the algorithm has been done using 99 responses which are treated as negative examples. Table 8.7 shows the classification of the code number 10 by the algorithm and both measures recall and precisian are acceptable hence the F measure is adequate.

**Table 8.7: Precision and Recall for code 10-Positive Naïve Bayes**

| | Relevant to code 10 | Not Relevant to code 10 | Total |
|---|---|---|---|
| Retrieved as Cluster 10 | 63 | 8 | 71 |
| Not Retrieved as Cluster 10 | 1 | 27 | 28 |
| Total | 64 | 35 | 99 |

| | |
|---|---|
| Precision | 0.690141 |
| Recall | 0.984375 |
| F | 0.811408 |

# Chapter 9: Discussion and Conclusion

## 9.1 Chapter overview

The final chapter elucidates the general discussion about the work carried out and explained in the previous chapters. Main focus in this section is to reveal the problem faced during the analysis and the pros and cons of the proposed algorithm in solving the problem of open-ended categorization. Future works and the suggestions to improve the proposed solution are presented at the bottom of the chapter.

## 9.2 Data Cleaning and Preprocessing

As mentioned earlier open-ended responses are unstructured texts with many typo errors and many symbols and numbers. Correcting such typo errors is vital task without removing the word completely. Manual and automatic processes are employed to correct the typo errors and remove the symbols. This task consuming considerable amount of time and cannot be fully automated.

Some open-ended responses are with more than a sentence. If a response contains more than two sentences then the response is split into several responses as the number of sentences are included in original response. The new responses are treated as separate responses from same respondent.

In open-ended response analysis the negation words such as “no”, “not”, “don’t”, “won’t” ...etc are important to measure the polarity of the responses. Therefore these are words are not removed by the process of stop word elimination. The words like “don’t” and “won’t” are normalized to the pattern to “do not” and “will not” prior to the remove the stop words.

The unique word list of non functional words is generated with the frequency of their occurrences. This list contains enormous set of words with the frequency starting form 1. To condense the calculation and algorithmic complexity in the proposed algorithm it is required to limit the size of the unique word list. The selection criteria have been created for this selection as to limit the words with frequency greater than or equal to 3. This criterion has been proved in the study by caring out several tests.

Some of these selected key words are with morphological variation of the stem of the words. It is believed that the Information contained in the responses may be lost and the result of the

proposed algorithm may be affected by removing the morphological variation. To test this argument the experiment is carried out with and without removing the morphological variation of the words in the unique word list. The same is done for the words in the tested responses as well. The result of the experiment shows that the performance of the attempt, by removing the morphological variation is much better than the other.

## 9.3 Hierarchical Clustering

The first proposed method of categorizing open-ended responses in this study the one of unsupervised method named as hierarchical clustering. As mentioned in the motivation of the study most of the open-ended responses are categorized and analyzed manually without using predefined codes. The unavailability of such predefined codes and a set of open-ended responses coded by the predefined codes as the training data, justifies the appropriateness of the unsupervised clustering method in analyzing the open-ended responses.

The clustering mechanism initiated by representing the open-ended response by a vector. The feature of the vector is a define term weight of the key words extracted from the responses. Hence the dimension of the vector is the cardinality of the key word list. One of the best weighting method for vector space model in information analysis is the tf-idf weight for given term. Therefore tf-idf value for a given key word is the weight of the vector in proposed hierarchical clustering for open-ended categorization. The weight tf-idf incorporates the key word frequency in the responses. Therefore more a word appears in the response the more it is estimated to be significant in this response. Idf measures how infrequent a word is in the responses. The idf value is estimated using the whole set of responses considered for categorization. It is clear that, if a key word is very frequent in the response collection, it is not considered to be particularly representative of this response. On the other hand, if the key word is infrequent in the response collection, it is believed to be, very relevant for the response as well. Many researchers have studied text categorization based on different term weighting schemes, but tf-idf scheme shows significant better performance than other weighting schemes (Lan et al., 2005). In this study we found the good performance of the tf-idf in the domain of open-ended categorization as well.

In hierarchical clustering the main parameter for the clustering is the linkage function. In this study the “average” linkage function is used to cluster the open-ended responses. The study is

extended to find the best linkage function over single, average and complete. The experiment in open-ended responses reveals that the best performer is average linkage over the other two. The similarity between two response vectors is measure by the cosine similarity. Among the similarity measures used in text categorization, cosine similarity is the simple and best measure to compare two documents.[19] The performance of Euclidian distance measure as the similarity of two open-ended responses for the categorization is poor compare to that of cosine similarity measure.

The number of cods for open-ended categorization is the most important measure to be decided by the study. The method is proposed using the tree structure constructed by the hierarchical clustering algorithm. The tree structure is represented by dendrogram. Cutting the branches off dendrogram at an appropriate level is the solution of getting the clusters and hence deciding the number of clusters. The process of cluster detection is referred to as tree cutting, branch cutting, or branch pruning (Huag A. , 2008). The constant height cutoff dendrogram method is implemented in the proposed solution thereby decided the number of codes. The method is flexible and simple for automation. Over 90% of the cases this is match with the number predefined manual codes in the tested sample.

The high F value in majority of the clusters shows the suitability of the hierarchical clustering in categorizing the open-ended responses in any research domain. Inadequate representation of responses for a particular code in the test data affects the clusters to be separated by others. The responses with same meaning but with different wording are clustered into different clusters as the semantic of common words are not considered for the algorithm. For an instance the responses containing “no Idea” and “do not know” are coded as two separate codes even though they semantically similar. The code can be merged by applying the latent semantic analysis for the responses in future studies.

## 9.4 Naïve Bayes Approach

Naïve Bayes classification technique is tested as the supervised method in categorization of the open-ended responses. Naïve Bayes is the classification procedure based on Bayes theorem. The best practice of conducting a statistical survey is conducting a pilot survey prior to the actual survey. A preliminary small survey conducted before the complete survey to test the effectiveness of the research methodology is the pilot survey. In the scenario of open-ended

analysis the possible code for the open-ended responses would be defined by this pilot survey. And a sample of the open-ended responses is categorized using these identified codes before coding the complete responses. The application of naïve bays is reasonable using the coded sample responses as the supervised training set.

In the classical Naïve Bayes, a 80-20% break up ratio between the training and testing data is suggested. But in the case of classifying open-ended responses this rule is not applicable since the training data set is small. Getting such labeled responses in statistical surveys are expensive and impracticable. However the study was carried out for testing the classical naïve bayes with this limitation. As in the hierarchical clustering approach the key word list was extracted and calculated the prior probabilities for selected key words. The accuracy of the open-ended response classification by the proposed method is far below the accepted level. The main reason for this poor performance may be the size of training data set. Further analysis of the misclassification is carried out and the significant categories that most respondent typed are considered for the analysis. The responses manually coded as number "2" were classified into two categories by the Naïve Bayes classification algorithm. The key words list constructed for these classified categories and compared.

**Table 9.1: Key word density classified as 2**

| Key Word | Frequency |
|---|---|
| energy | 14 |
| carbon | 10 |
| trust | 9 |
| save | 12 |
| use | 6 |
| looks | 2 |
| impact | 2 |
| environment | 2 |
| minimizing | 2 |

**Table 9.2: Key word density classified as 1**

| Key Word | Frequency |
|---|---|
| energy | 14 |
| carbon | 10 |
| reduce | 6 |
| trust | 5 |
| save | 8 |
| offsetting | 3 |
| footprint | 3 |
| use | 4 |
| general | 2 |
| second | 2 |
| concerned | 2 |
| nuclear | 2 |
| advises | 2 |
| est | 2 |
| trying | 2 |

Comparing the above key word lists in table 9.1 and 9.2 reveals that the responses contain key words “energy” and “carbon” which are coded as code number 2 in the manual categorization but in algorithmic categorization these are further divided. The word “reduce” is frequent in the second set which is coded as 1 in algorithmic classification. There are 21 responses has been categorized as 1 by the naïve bayes classification. In these 21 responses the word “reduce” or its derivative is present.

## 9.5 Positive Naïve Bayes Approach

It is believed that the reason for not achieving the expected accuracy by the classical Naïve Bayes is the inadequate training sample size. As mentioned in earlier the solution is incremental learning or positive Naïve Bayes classification. The analysis shown in chapter 8 section 8.5, the responses with short sentence were classified with high accuracy whereas the long sentence responses were classified to different categories. Since the method is totally based on the key word density of the responses the classification mislead in responses with high key word cloud.

**Table 9.3: Key word density classified as 2**

| Key word | Frequency |
|---|---|
| energy | 17 |
| save | 14 |
| carbon | 12 |
| trust | 9 |
| use | 5 |
| looks | 3 |
| environment | 3 |
| reduce | 3 |
| minimizing | 2 |
| impact | 2 |
| concentrates | 2 |

**Table9.4: Key word density classified as 2**

| Key word | Frequency |
|---|---|
| energy | 8 |
| carbon | 7 |
| save | 6 |
| trust | 3 |
| concerned | 3 |
| reduce | 2 |
| nuclear | 2 |
| est | 2 |
| emissions | 2 |

Even though there is not much of a difference in the key words for the two groups, the responses have been categorized into different clusters.

## 9.6 Comparison of Hierarchical and Naïve Bayes methods

Results reveal that Naïve Bayes approach outperforms than the hierarchical method. However, there are some prerequisite needed to apply the Naïve Bayes. A set of labeled open ended

responses are the main requirement to run the naïve bayes. Hierarchical clustering applied to the all the responses available at once. But the number of codes is an essential requirement to decide the clusters.

## 9.7 Final Conclusion and Future work

Hierarchical method is preferred when there is no labeled data for the supervised training algorithms. The proposed hierarchical algorithm shows significant results in coding the open-ended responses. Some clusters retrieved by the algorithm contain several similar words as described in section 7.3.5, these words can be merged by considering the similarity of meaning. The latent semantic indexing algorithm is most popular technique of combining such similar words to one cluster. Therefore latent semantic indexing is proposed as the future work to the study. Another future work will be wiring these similar words using WordNet into a single entity by considering the synsets [WWW 03].

When the fraction of coded responses is available the supervised categorization algorithm is appropriate to code the open ended responses. But the experiment revealed that the performance of such supervised algorithm is not adequate to code the open ended responses. As described earlier this is due to lack f training data. As alternative to the naïve bayes supervised algorithm for open ended response categorization, a positive naïve bayes showed drastic improvement over the classical naïve bayes.

The two methods, hierarchical and the supervised categorization method can be combined to have hybrid solution in future. The coded responses by the hierarchical method can be input to the supervised techniques as the training data set. This will reduce the cost of crating the label data set to the supervised method.

The both algorithms proposed in the study required the electronically typed responses. At present many surveys are conduct online hence responses to all open ended questions are available electronically. An online open ended response categorization tool can be proposed as a future work by combining the two algorithms invented in the research.

Responses with sentiment are more interesting and valuable in opinion extracting systems. Sentiment is the polarity of the open ended response or opinion. Polarity of the responses can be either positive or negative and further a value between both. An application is proposed as a future work to add the sentiment to the coded responses. This can be easily integrated to the

proposed algorithm since the open ended responses were already grouped in some criteria automatically by the algorithm.

## References.


1. Brown P. F. ,Pietra V.J.D, Desouza P.V. Lai J.C. and Mercer R. L (1992). Class Based n-gram Models for Natural Languages. Computational Linguistics Volume 18 Issue 4, December 1992 MIT Press Cambridge, MA, USA

2. Bullington Jim , I. Endres, and M. A. Rahman (1998). *Open-Ended Question Classification Using Support Vector Machines*, Department of Computer Science, University of West Georgia.

3. Giorgetti Daniela, Sebastiani Fabrizio (2003). Automating Survey Coding by Multiclass Text Categorization Techniques, Journal of the American Society for Information Science and Technology .

4. Kang M, K. Asakimori, A.Utsuki and M.Kaburagi, (2005). Automated Text Clustering on Responses to Open-ended Questions in Course Evaluations, ITHET 6th Annual International Conference.

5. Hiramatsu A, S.Tamura, H.Oiso, N.Komoda, (2005). The study of method of a typical opinion extraction from answers in open-ended answers, IEEJ Transactions on Electronics, Information and Systems Vol. 125

6. Giorgetti.D and F. Sebastiani (2000). Automating Survey Coding by Multiclass Text Categorization Techniques, Istituto di Linguistica Computazionale Consiglio Nazionale delle Ricerche, Italy.

7. Miller, G. and W.G. Charles (1991). "Contextual Correlates of Semantic Similarity", Language and Cognitive Processes, Vol. 6, No. 1, 1-28.

8. Jain A. K, Murty M.N, Flynn P.J (1999). Data Clustering: Review, ACM Computing Surveys

9. Johnson R.A, and D.E.Wichern, (1996). Applied Multivariate statistical Analysis, Third Edition

10. Manning C. & Schütze H.(1999). Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA:

11. Kwok J.T. (1998). Automated Text Categorization Using Support Vector Machine, In Proceedings of the International Conference on Neural Information Processing (ICONIP).

12. Han E.H., Karypis G., Kumar V. (1999) Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification.

13. Murphy K. P. (2006) Naive Bayes classifiers, University of British Columbia.

14. Sun A., Lim E.(2001) Hierarchical Text Classification and Evaluation, Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference.

15. Ghwanmeh S. H. (1998). Applying Clustering of Hierarchical K-means-like Algorithm on Arabic Language, International Journal of Information Technology Volume 3 Number 3,1998.

16. Denis F., Gillerom R., Tommasi M. (2002). Text Classification from Positive and Unlabeled examples. In Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2002).

17. Lan M., Tan C., Low H.(2005). A Comprehensive Comparative Study on Term Weighting Schemes for Text Categorization with Support Vector Machines(2005) In Posters Proc. 14th International World Wide Web Conference.

18. Huang A. (2008). Similarity Measures for Text Document Clustering, Proceedings of the Sixth New Zealand Computer Science Research Student Conference NZCSRSC2008 Christchurch New Zealand.

19. Langfelder P., Zhang B. and Horvath S.: Defining clusters from a hierarchical cluster tree (http://bioinformatics.oxfordjournals.org/content/24/5/719.full)

20. Manning C. D, Raghava P. and Schütze H. (2009) Introduction to Information Retrieval (2009), Cambridge University Press.

21. [WWW 01] http://en.wikipedia.org/wiki/Course_in_General_Linguistics/ [07/04/2011]

22. [WWW 02] www.vocabulary.com/definition/categorization/ [17/09/2011]

23. [WWW 03] www. http://wordnet.princeton.edu/ [26/11/2012]