2007

# Summaritive Digest for Large Document Repositories with Application to E-Rulemaking

Lijun Chen
*Wright State University*

# SUMMARITIVE DIGEST FOR LARGE DOCUMENT REPOSITORIES WITH APPLICATION TO E-RULEMAKING

A dissertation submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

By

LIJUN CHEN
M.S., University of Dayton, 2001

---

2007
Wright State University

WRIGHT STATE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

August 17, 2007

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Lijun Chen ENTITLED Summaritive Digest for Large Document Repositories With Application To e-Rulemaking BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF Doctor of Philosophy.

Guozhu Dong, Ph.D.
Dissertation Director

Thomas Sudkamp, Ph.D.
Director, Ph.D. Program of CS&E

Joseph F. Thomas Jr., Ph.D.
Dean, School of Graduate Studies

Committee on Final Examination

Guozhu Dong, Ph.D.

Mateen Rizki, Ph.D.

Soon Chung, Ph.D.

Lop-Fat Ho, Ph.D.

Jennifer Seitzer, Ph.D.

# Abstract

Large document repositories need to be organized and summarized to make them more accessible and understandable. Such needs exist in many applications, including web search, e-rulemaking (electronic rulemaking) and document archiving. Even though much has been done in the areas of document clustering and summarization, there are still many new challenges and issues that need to be addressed as the repositories become larger, more prevalent and dynamic. In this dissertation, we investigate more informative ways to organize and summarize large document repositories, especially e-rulemaking feedback repositories (ERFRs), so that the large repositories can be managed and digested more efficiently and effectively. Specifically, we mainly consider the following four tasks: 1) identifying important aspects of ERFR, 2) constructing cluster descriptions for document clustering, 3) clustering of ERFR with simultaneous construction of succinct cluster descriptions, and 4) selecting representative arguments for ERFR clustering.

We propose to organize and summarize e-rulemaking feedbacks based on three different major aspects of the rulemaking process, in order to meet the different needs of the rule-writers or analysts; the three aspects are: opinions (O), issues (I) and stakeholders (S). We introduce an *OIS-based* approach to producing informative summaritive digest (SD) for given ERFRs. In addition, several novel concepts, approaches and algorithms are introduced, including the CDD measure, *active feature selection* (AFS), Pagoda search algorithms, etc.

An SD, simply put, consists of a document clustering, along with certain succinct cluster descriptions (SCDs) and representative arguments (RAs) for each cluster in the clustering. The clustering of an SD can be constructed in either a flat or hierarchical manner. For hierarchical clustering, each level of the hierarchy can be constructed by emphasizing one of the O, I, and S aspects. Different orders of O, I and S can be used for the levels of the hierarchy. Different clusterings could be used to meet the needs of different users. Given a goodness measure, a "best" clustering can be recommended to the user. An SCD consists of a set of carefully selected terms along with some statistics, and the RAs are some typical *arguments* selected from each cluster. An RA should be a statement where certain major stakeholders have expressed opinions on some of the important issues. Collectively, an SD provides an informative navigation aid for the rule-writers and analysts to manage and digest large ERFRs.

We conduct an experimental evaluation on our approaches by using some publicly available ERFRs. The results suggest that the SD not only helps user for "browsing" the feedbacks, but also gives the users some high-level sense about the feedbacks before they dig into each individual comment. The results also show that our approaches are efficient and scalable for managing large document repositories.

Even though we devoted special attention to the application of e-rulemaking, we believe that most of the ideas are very generic and can be easily applied to other types of repositories, including digital archives.

# Contents

# List of Figures

# List of Tables

# Acknowledgement

Because of my personal desire for this doctoral degree, the debts that I have accumulated during my study years at Wright State are numerous. I would like to thank all the people who have provided me with assistance over the years. All of your support is greatly appreciated.

First of all, I would like to thank my beloved parents and family: my wife Chunxia, son Eric and daughter Anna. Without their support and encouragement I never would have made it to this point. Their understanding and backing helped sustain me through the ups and downs of conducting research works. I feel very lucky to have such a wonderful supportive family.

My deepest thanks go to my principal dissertation advisor, Professor Guozhu Dong, for his patient guidance, constant encouragement and generous support. His dedication to helping students identify and pursue their research interests has made this dissertation possible. He is very knowledgeable and full of ideas on this research topic. Over the past few years I have learned a tremendous amount from him about both research and life, and I am grateful to have had the opportunity to work with him.

Many thanks to Professors Mateen Rizki, Soon Chung, Lop-Fat Ho and Jennifer Seitzer for their advice and support throughout this research project. My research is an interdisciplinary work, and needs knowledge from the areas of database, artificial intelligence, information retrieval, statistics and other computer science

subareas to complete. Each professor has provided significant and unique feedbacks in their respective areas of expertise that helped this research to come together.

I would like to thank Professors James Buckley, Jennifer Seitzer and Barbara Smith for their mentoring and support when I studied at University of Dayton, and for their encouragement to pursuing this doctoral degree. Thanks also go to all the members of the Data Mining Research Lab, especially Shihong Mao, Chunyu Jiang and Ying Sun. I will treasure those wonderful times when we had helpful discussion, humorous talking, and hilarious laughing in the lab.

I would also like to acknowledge the generous support provided by the Dayton Area Graduate Studies Institute (DAGSI) scholarship over the years.

Finally, I apologize to anyone whom I've forgotten to thank here.

# Chapter 1

# Introduction

We will start by briefly giving the motivation of this research project. Then we will highlight some related approaches, followed by our research goals. The outline of this dissertation will be given at the end of this chapter.

## 1.1 Motivation

In this networked digital era, most individuals, companies and government agencies are faced with large repositories of documents which may consist of emails, news, reports, proposals, regulations, etc. These repositories can be collected passively or actively from the Internet, or built as results of digital libraries, digital governments, or digital archives. Clearly, both the types and sizes of such repositories will continue to expand, and will become larger, more prevalent and dynamic.

With the rapid increase of available documents everywhere, the challenge for managing and digesting them has also been increasing dramatically. In some areas, for example web searching, the amount of available information is overwhelming for the majority of the users. This phenomenon has often been referred to as information overload [118]. Therefore, there is an urgent need for the research

community to come up with better ways to organize, summarize and label those repositories so that they can be used effectively and efficiently. Obviously, without good ways to navigate and digest them, we cannot get out of the "information rich and knowledge poor" situation.

This research was primarily motivated by the needs discussed above. Besides generic document repositories, such needs are also taking place from digesting e-rulemaking feedback repositories. We will briefly discuss both the generic situation and the e-rulemaking application below.

### 1.1.1 Organizing Large Document Repositories

For large document repository, it is very important to find better ways to organize the documents based on their content. More importantly, it is very desirable to have informative and succinct descriptions for the document clusters. With good clustering results and succinct cluster descriptions, users can easily get a high-level sense of what the document repository contains. Therefore, a good document clustering with good cluster descriptions can be a useful handle for users to digest the underlying documents.

For the past several decades, there has been much research on organizing document collections. Most recently, there has also been much research on document summarization. However, with more and more documents available daily, the need for more effective ways to organize and summarize them becomes more urgent. In addition, because of the high dimensionality of the documents and the heterogeneous nature of document types, the problem is becoming increasingly more challenging.

### 1.1.2 Digesting E-Rulemaking Feedbacks

Every year thousands of government personnel at over 150 federal agencies and sub-agencies collaborate with stakeholder interest groups, lobbyists, lawyers, and citizens to craft as many as 8,000 regulations [67]. For each proposed rule/regulation, there could be many (e.g. millions of) feedbacks from the public in different forms, such as formal letter, email, fax, etc. For such large amount of text data, it is a great challenge for the rule-writers to manage and digest them efficiently and effectively.

E-rulemaking can benefit greatly from the research results in text mining. However, as the e-rulemaking process becomes more prevailing in the recent years, there are more challenges that need to be addressed. In particular, when faced with large amount of feedbacks, rule-writers want to have some effective ways to know 1) what are the important issues that the public is concerned with? 2) which groups (i.e. stakeholders) are more concerned about these issues? 3) what are their opinions? 4) what are the typical arguments to support their opinions? etc. Therefore, it is highly desirable to organize the feedbacks based on some identified aspects/dimensions, in order to help the rule-writers or analyst to digest the feedbacks more easily.

## 1.2 Related Approaches

The research communities have been working on the problem of organizing and summarizing large document repositories for the past several decades. With the emerging applications of e-government, such as digital archiving and e-rulemaking, research in this area has been re-energized.

### 1.2.1 Handling Generic Document Repositories

There have been many studies on "understanding" documents from different aspects [28]. Such efforts have often been collectively referred to as text mining. From the perspective of IT professionals, some of the most effective approaches for handling large document repositories are document classification, clustering and summarization.

- Classification. This is a supervised approach, in which documents are classified into predefined taxonomies, similar to the organization of Yahoo [130] categories and DMOZ [25] directories. Some of the popular classification approaches are K-NN, EM, SVD, etc. [48].

- Clustering. This is an unsupervised (semi-supervised) approach, in which documents are automatically grouped into auto-determined (or predefined) number of groups based on their content. Document clustering has attracted a great deal of attention in recent years. Some new clustering algorithms have been developed, and some old methods have been updated and reworked for documents. A survey on document clustering can be found in [15, 8].

- Summarization. Document summarization is a very active research topic [119, 28], and it can be for a single document or multiple documents. A summary can be either an extract (consisting entirely of material from original input) or an abstract (at least some of the material is not present in the input) [84]. Document summarization can be roughly categorized into the following approaches: sentence based, term based and template based. We will provide more details on these in chapter 2.

There are also many studies on document visualization, indexing, storage compression, etc., which are not major concerns of this dissertation.

### 1.2.2 Handling E-Rulemaking Feedbacks

Most of the techniques used to manage generic document repositories can be directly applied to the e-rulemaking feedbacks. However, there are still other unique challenges that need to be addressed for e-rulemaking feedbacks, such as new social, political, and technical challenges.

Recently, there is a growing interest in e-rulemaking research, which brings together different communities interested in various aspects of e-rulemaking feedbacks, such as IT professionals, social scientists and government officials. Some of the active research groups are [122], [14] and [49]. Some test datasets have been made available at [60].

Some IT challenges that have received attention recently are near-duplicate detection [131], cluster labels [120, 17], rule relatedness analysis [71] and multi-aspect analysis of text [67].

## 1.3 Research Goals

As more and more document repositories become available, the needs for better ways to organize and digest them are pressing. In this dissertation, we will study methods to meet those needs and to address new challenges. In addition to considering these problems in the generic setting, we will pay more attention to the specific application of e-rulemaking.

Our research will mainly focus on producing informative summaritive digest (SD) for large E-Rulemaking Feedback Repositories (ERFRs). An SD, simply put, is a document clustering with succinct descriptions (SCDs) and representative arguments (RAs) for each cluster. With the organized structure and descriptive information associated with it, the SD can be treated as a "virtual map" for the given repository to improve the comprehensibility and usability of the underlying

documents. Therefore, the SD is not only a clustering with summarization but also a navigation aid for the give document repository.

In this dissertation, we also try to utilize the domain characteristics to boost the quality and usefulness of the SDs. For example, stakeholders, issues and opinions are probably common concerns for all proposed rules. Moreover, each proposed rule can be used as background knowledge when producing the SD.

More specifically, we plan to consider the following four tasks:

- Identifying important aspects of ERFR
- Constructing cluster descriptions for document clustering
- Clustering of ERFR with simultaneous construction of succinct cluster descriptions
- Selecting representative arguments for ERFR clustering

While there are some overlaps among those tasks, there are unique challenges and different emphasis for each one of them.

In addition, we also consider important issues such as how to evaluate the quality of the SDs (including both clustering and summarization qualities).

## 1.4 Dissertation Outline

The rest of this dissertation is organized into the following six chapters:

- Chapter 2 gives some preliminaries on the techniques and terminologies that will be used throughout this dissertation. These include e-rulemaking, document clustering, document summarization, part-of-speech (POS) tagging and WordNet.

- Chapter 3 covers the approaches for identifying important aspects of e-rulemaking, which are stakeholders, issues and opinions, for organizing ERFR.
- Chapter 4 describes how to construct cluster descriptions (CDs), which consist of a set of terms, for any given document clustering.
- Chapter 5 presents approaches that perform clustering and construct succinct cluster descriptions (SCDs) simultaneously for any given ERFR.
- Chapter 6 discusses how to select representative arguments (RAs) for given ERFR clustering.
- Chapter 7 summarizes the contributions of this dissertation, and examines areas that are important for future research work.

We note that experiments are reported in each of Chapters 3 to 6 to evaluate the approaches of those chapters.

# Chapter 2

# Preliminaries

In this chapter, we will give some preliminaries on the techniques and terminologies that will be used throughout this dissertation. These include electronic rulemaking (e-rulemaking), document clustering, document summarization and some related works in those areas. In addition, we will also briefly discuss part-of-speech (POS) tagging and WordNet at the end of this chapter.

## 2.1 E-Rulemaking (Electronic Rulemaking)

### 2.1.1 Introduction

Electronic rulemaking, or e-rulemaking, refers to the use of digital technologies by government agencies in the rulemaking process [49]. Rulemaking procedures are defined by law in Section 553 of the Administrative Procedure Act of 1946, which requires agencies to publish a notice of proposed rulemaking in the Federal Register; to permit any interested party to engage in the rulemaking process through provision of written data, views, or arguments; and to publish the rule 30 days before it takes effect [34]. In order to issue a rule, a regulatory agency must 1) publish a notice; 2) collect public comments; and 3) incorporate the feedbacks and


Figure 2.1: Notice-and-Comment Rulemaking

make the final rule. This procedure is known as "notice-and-comment" rulemaking [20]; see Figure 2.1 for an illustration.

In the past, public comment was submitted to the U.S. federal government primarily in paper form. However during the last several years the government has begun to allow comments to be submitted electronically in some cases. Recently a new Regulations.gov [103] web site was created to make it easier for citizens to examine and comment on proposed regulations, so the volume of electronic comments is expected to grow rapidly.

### 2.1.2 Challenges

E-Rulemaking offers opportunities for the government to reduce its costs and improve the quality of notice-and-comment rulemaking. However it also poses a variety of new social, political, and technical challenges. Some of the research opportunities in the IT area are text clustering, information retrieval, near-duplicate detection, opinion identification, summarization, etc. [20]. There are some active research groups working on e-rulemaking, such as University of Pittsburgh [122], CMU [14] and Harvard [49].

The following are the findings from [112]: a future e-rulemaking system will require more interactive features; priorities must be set among desired features; background and training focused on the rulemaking process itself are needed for

successful widespread implementation; cross-agency capabilities are needed; and there is a need to be able to access previous rulemaking dockets. Much research has been done to help facilitate this process, such as text clustering [59, 15, 115, 8], term extraction [11], syntactical constituent identification [5, 9], identifying subjective expressions [126, 127], distinguishing facts from comments [134] and sentiment classification [96, 50, 121].

However, there are still some new challenges in the e-rulemaking process that need to be addressed. One of the challenging problems is how to analyze and categorize text according to several novel dimensions, such as stakeholder, opinions and argument [110, 109, 111]. In this dissertation, we will study ways to address those challenges.

### 2.1.3 Recent Development

Researchers at USC ISI and CMU are developing a new text processing tool that can help perform advanced analysis of large collection of text commentary [108]. This is a three-year project which started in October 2004 and it has been funded under the National Science Foundation's Digital Government program. The focuses of the project are on text clustering, text searching using information retrieval, near-duplicate detection, opinion identification, stakeholder characterization, and extractive summarization. There are some results about near-duplicate detection have been reported in [131].

References [69, 70] proposed an information infrastructure for regulation analysis, which includes a document repository and tools for compliance assistance and similarity analysis. A regulatory repository is developed based on an XML format, the tree hierarchy of regulations and its referential structure are preserved by properly structuring XML elements. The main application of this work is to help identify related draft provisions and public comments. Relatedness analysis is

performed by comparing the extracted features as well as structural and referential information from regulations [72, 71]. Based on the approach in [70], feedbacks can be organized using the hierarchical structure of rule labels. For example, a rule is usually presented by sections and subsection, such as 1.1, 1.1.1, 1.1.2, 1.2, 2.1, etc. Those section and subsection numbers form a hierarchy.

References [120] and [17] try to produce succinct cluster labels for given document clustering. Their results showed that such labeling efforts can help the users better to understand and organize large collection of document, such as e-rulemaking feedbacks.

Reference [67] tries to analyze each document based on different aspects of text, such as argument structure, topics, and opinions. The main focus was on the analysis of each feedback. It showed that such multidimensional text analysis could help highlight the main focus of each feedback.

### 2.1.4 Available Data Sets

Some test datasets have been made available at [60] for the research communities. A brief description of these data sets is given below. We will use these datasets in this dissertation.

- *EPA-CWA*: feedbacks from the U.S. Environmental Protection Agency (EPA) about revising the Clean Water Act (CWA). There are about 500 comments (extracted from .pdf files) distributed in a zip file.

- *EPA-NESHAP*: feedbacks from the EPA about setting National Emission Standards for Hazardous Air Pollutants (NESHAP). There are two sample files, and each contains about 1000 comments.

- *DoT-CAFE*: feedbacks from the U.S. Department of Transportation (DoT) about revising Corporate Average Fuel Economy (CAFE) standards. There

are 1000 randomly selected comments distributed in a tar file.

- *DoA-SWPM*: feedbacks from the U.S. Department of Agriculture (DoA) about importation of Solid Wood Packing Material (SWPM) standards. There are 956 public comments distributed in a text file.

- *DoA-NOP*: feedbacks from the U.S. Department of Agriculture's National Organic Program (NOP). There are totally 20936 comments distributed in 3 volumes. ■

## 2.2 Document Clustering

Clustering is the process of grouping the data into clusters so that objects in each cluster have high intra-cluster similarities and objects in different clusters have low inter-clusters similarities [64].

Document clustering is an unsupervised learning process in which documents are automatically grouped together based on their contents, so that documents are very similar within a cluster and dissimilar from other clusters.

Document clustering has attracted a great deal of attention in recent years. Some new clustering algorithms have been developed, and some old methods have been updated and reworked for documents. A survey on document clustering can be found in [15, 8].

In the following subsections, we will briefly discuss the challenges faced in document clustering, and the common techniques and procedures used in document clustering. We will also describe some well-known document clustering algorithms.

### 2.2.1 Challenges

There are many challenges faced by document clustering because of the characteristics of the document data type.

- High Dimensionality. The biggest challenge for document clustering is the "curse of dimensionality". In general terms, problems with high dimensionality result from the fact that a fixed number of data points become increasingly "sparse" as the dimensionality increase [114]. This is the case for large document collections since the number of unique words is usually very large.

- Heterogenous. Document repositories may contain documents that are from different sources and have different formats and purposes. For example, the proposed e-regulations are in formal witting, and often come from government in PDF or HTML format. On the other hand, feedbacks from public may contain a lot of informal languages, and could be coming in any format.

- Fast growing and dynamic. In this networked digital era, documents created by individuals, companies and government agencies are growing in a very fast pace. In addition, even the "same" document may change everyday, such as web site.

- Lack of informative cluster descriptions. Documents are different from numeric data since documents (and words contained in document) have meanings. As the ever-growing of large document repositories, it is highly desired to have succinct, yet informative, description for each document cluster so that users can quickly get a high-level sense of underling documents. ■

### 2.2.2 Document Preprocessing

The first step, also a critical step, for text mining is the pre-processing phase, consisting of a number of complex tasks aimed at making documents "machine readable" and eliminating "noises".

Some of the commonly used steps for documents preprocessing are:

- Tokenization: Tokenization is the process of mapping sentences from character strings into strings of words. For example, the sentence "This is a 3-year project" can be tokenized into following tokens: "This", "is", "a", "3-year" and "project".

- Data cleaning: Usually any non-textual information, such as HTML tags, punctuation marks and digits, are removed from the documents.

- Stopwords removal: Stopwords are typical frequently occurring words that have little or no discriminating power, such as "a", "about", "all", etc., or other domain-dependent words. Stopwords are often removed.

- Stemming: Typically, the stemming process will be performed so that the words are transformed into their root form. For example, "cluster", "clusters" and "clustering" are all transformed to "cluster". Among all the stemming algorithms, such as Porter's stemmer [99], Paice/Husk's stemmer [95], etc., Porter's algorithm is the most commonly used one.

- Others: More often, some feature extraction or selection techniques are used so that less discriminating terms are removed to reduce the dimensionality. For example, all words that occur in less than five documents are removed.

■

### 2.2.3 Document Representation

In order to reduce the dimensionality of the documents and make them easier to handle, the documents have to be transformed into certain format which describes the contents of the documents. The Vector-Space Model (or Bag of Words) is the most commonly used model in the information retrieval community.

In the simplest form of Vector-Space Model, each document $d$ is represented by the term-frequency ($tf$) vector $d_{tf} = (tf_1, tf_2, \ldots, tf_m)$, where $tf_i$ is the frequency of the $i_{th}$ term in $d$. Often, the document vectors are normalized to unit length to allow comparison of documents with different lengths. Note that a term may either be a single word or consist of several words.

Currently, the TFxIDF (Term Frequency times Inverse Document Frequency) model is the most popular model. In the TFxIDF model, a term $t_i$ is weighted by its frequency in the document $tf_i$ times the logarithm of its inverse document frequency, i.e., $tf_i * log_2(N/n)$, where $N$ is the number of documents in the collection and $n$ is the number of documents where the term $t_i$ occurs at least once. Note that by using the TFxIDF model, terms that appear too rarely or too frequently are ranked lower than the other terms.

The Latent Semantic Indexing (LSI) model is a special case of Vector-Space Models [74]. In the LSI model, terms and documents are represented by an incidence matrix $A$. Each of the $M$ unique terms in the document collection are assigned a row in the matrix, while each of the $N$ documents in the collection are assigned a column in the matrix. Since the number of terms in a given document is typically far less than the number of terms in the entire document collection, the matrix $A$ is usually very sparse. The LSI model often uses the Singular Value Decomposition (SVD) [40] technique to reduce the dimensionality of the term-document space.

### 2.2.4 Similarity Measures

All clustering algorithms are based on certain similarity measures. The most commonly used similarity measure is the cosine measure when documents are represented in the vector-space model.

Given two document vectors $d_i$ and $d_j$, the cosine measure is defined by the cosine of the angle between the two vectors:

$$sim_{cos}(d_i, d_j) = \frac{d_i \centerdot d_j}{\|d_i\| * \|d_j\|}$$

where $\centerdot$ denoted the vector dot product and $\| \ \|$ denotes the length of a given vector. If the document vectors are of unit length, the above formula simplifies to $sim_{cos}(d_i, d_j) = d_i \centerdot d_j$.

There are also some other similarity measures, such as Minkowski distance (especially, the Enclidean distance), Pearson correlation, extended Jaccard coefficient, mutual neighbor, etc. [64, 116].

### 2.2.5 Document Clustering Approaches

We now discuss some well-studied document clustering algorithms. Most of the existing approaches to document clustering are based on either probabilistic methods, or distance and similarity measures [35]. There are various ways to categorize clustering algorithms. For example, one way is the one mentioned in [48] where clustering methods are categorized as partitioning, hierarchical, density-based, grid-based, model-based and etc. Another way is the one described in [42], such as hierarchical vs. partitional clustering, soft vs. hard clustering, or parametric vs. non-parametric clustering. In this survey, we just group those clustering methods based on their salient characteristics.

#### 2.2.5.1 Hierarchical Methods

One popular approach in document clustering is *agglomerative hierarchical clustering* [64]. In this approach, a hierarchy is built bottom-up by iteratively computing the similarity between all pairs of clusters and then merging the most similar pair. Different variations may employ different similarity measuring schemes [138]. Reference [115] shows that Unweighted Pair Group Method with Arithmetic Mean (UPGMA) is the most accurate one in its category. The hierarchy can also be built top-down which is known as the *divisive hierarchical clustering*. It starts with all the documents in the same cluster and iteratively splits a cluster into smaller clusters until a certain termination condition is fulfilled.

The hierarchical algorithms usually suffer from their inability to perform adjustment once a merge or split has been performed. This inflexibility often lowers the clustering accuracy. Even the most accurate one in the category, UPGMA, is not scalable for handling large data sets in document clustering as experimentally demonstrated in [37], due to the complexity of computing the similarity between every pair of clusters.

#### 2.2.5.2 Partitioning Methods

K-means and its variants [64, 52, 23, 24] represent the category of partitioning clustering algorithms that create a flat, non-hierarchical clustering consisting of $k$ clusters. The k-means algorithm iteratively refines a randomly chosen set of $k$ initial centroids, minimizing the average distance (i.e., maximizing the similarity) of documents to their closest (most similar) centroid. The *bisecting k-means* algorithm first selects a cluster to split, and then employs basic k-means to create two sub-clusters, repeating these two steps until the desired number $k$ of clusters is reached. Reference [115] shows that the bisecting k-means algorithm outperforms basic k-means as well as agglomerative hierarchical clustering in terms of accuracy

and efficiency.

Both the basic and the bisecting k-means algorithms are relatively efficient and scalable, and their complexity is linear in the number of documents [68]. As they are easy to implement, they are widely used in different clustering applications. A major disadvantage of k-means, however, is that an incorrect estimation of the input parameter, the number of clusters, may lead to poor clustering accuracy. Also, the k-means algorithm is sensitive to noise that may have a significant influence on the cluster centroid, which in turn lowers the clustering accuracy [37]. The k-medoid algorithm [64] was proposed to address the noise problem, but this algorithm is computationally much more expensive and does not scale well to large document sets.

#### 2.2.5.3 Graph Based Methods

One special type of partitioning algorithm is to transform the problem so that graph theory can be used [44, 116]. In this approach, the objects (documents or words) to be clustered can be viewed as a set of vertices. Two vertices are connected with an undirected edge of positive weight based on certain measurement.

In the clustering process, a set of edges, called edge separator, are removed so that the graph is partitioned into $k$ pair-wise disjoint sub-graphs. One objective of the partitioning is to find such separator with a minimum sum of edge weights. Another objective is to keep the numbers of objects in the clusters approximately equal. With those particular constraints, the graph partitioning algorithm is usually NP-hard [116].

#### 2.2.5.4 Frequent Itemset Based Methods

Reference [125] introduced a new criterion for clustering transactions using frequent itemsets. The intuition of this criterion is that many frequent itemsets

should be shared within a cluster while different clusters should have more or less different frequent itemsets. By treating a document as a transaction and a term as an item, this method can be applied to document clustering. The Hierarchical Frequent Term-based Clustering (HFTC) method proposed by [7] attempts to create a hierarchy of clusters by using the notion of frequent itemsets. HFTC greedily selects the next frequent itemset, which represents the next cluster, minimizing the overlap of clusters in terms of shared documents. The clustering result depends on the order of selected itemsets, which in turn depends on the greedy heuristic used. Although HFTC is comparable to bisecting k-means in terms of clustering accuracy, experiments show that HFTC is not scalable [37].

#### 2.2.5.5 Conceptual Clustering

Conceptual clustering is a special type of document clustering. In conceptual clustering, a group of objects forms a class only if it is descried by a *concept*. Conceptual clustering consists of two components: (1) it discovers the appropriate classes having high intra-class similarity and low inter-class similarity, and (2) it forms descriptions for each class, as in classification [48].

*COBWEB* [33] is a well-know incremental conceptual clustering algorithm in AI community. COBWEB works incrementally, it updates the clustering instance by instance using the *category utility function*. Each cluster is also summarized by a list of attributes and associated probabilities. COBWEB can create a tree-like clustering, with leaves representing each instance in the tree, the root representing the entire data set, and branches representing all the clusters and subclusters within the tree.

ITERATE [10] is another conceptual clustering algorithm. It employs: (i) a data ordering scheme and (ii) an iterative redistribution operator to produce maximally cohesive and distinct clusters.

The attribute-oriented induction [46, 47] method integrates a machine learning paradigm, especially learning-from-examples techniques, with set-oriented database operations and extracts generalized data from actual data in databases. This method summarizes the information in a relational database by repeatedly replacing specific attribute values with more general concepts according to user-defined concept hierarchies, or by forming the more general concepts on the fly.

#### 2.2.5.6 Others Methods

Some of the other clustering methods are:

**SOM Methods.** The Self-Organizing Map (SOM) is a clustering method with roots in Artificial Neural Networks (ANN). SOM is based on competitive learning (winner-takes-all), and it is often referred to as a topographic map [51]. SOMs have been used extensively for two purposes: cluster analysis [51] and visualization [104]. Although SOM is efficient and simple to implement, studies suggest that it typically performs worse than the other techniques such as k-means [80].

**Information Bottleneck Methods.** In the information bottleneck method, relevant information in a signal $x \in X$ is used to provide information about another signal $y \in Y$ [117]. Reference [113] uses the information bottleneck method to cluster documents by using word clusters.

**FCA Based Methods.** Formal Concept Analysis (FCA) uses order theory to analyze the relationship between objects $G$ and their features $M$. FCA identifies from such a data description, a so called formal context $K$, a set of features $B \subseteq M$ which are correlated with a set of objects $A \subseteq G$. Such a correlated pair is called a formal concept $(A, B)$ [54, 19]. FCA based clustering method is very expensive. However, the clustering organization is a lattice, rather than a hierarchy [18].

**BEA-Partition Methods.** Bond Energy Algorithm (BEA) is often used in psychology and database design. Reference [80] uses a BEA-partition method to clus-

ter genes based on extracted functional keywords among genes from MEDLINE abstracts.

### 2.2.6 Clustering Quality Measures

The quality of document clustering is often measured by using the amount of difference between the "natural" and the algorithm generated clusters.

A widely used quality measure for clustering and information retrieval methods is F-score [123], also called *F-measure*.

Suppose the original clustering is $\mathcal{C} = \{C_1, ..., C_L\}$, and the algorithm generated clustering is $\mathcal{C}' = \{C'_1, ..., C'_L\}$. For each $i$, the F-score for $C'_i$ and $C_i$, denoted by $F(C'_i, C_i)$, is defined as

$$F(C'_i, C_i) = \frac{2 * P(C'_i, C_i) * R(C'_i, C_i)}{P(C'_i, C_i) + R(C'_i, C_i)},$$

where $P(C'_i, C_i) = |C'_i \cap C_i|/|C_i|$ is the *precision*, which represents the percentage documents assigned by algorithm that are in fact belong to the original cluster; and $R(C'_i, C_i) = |C'_i \cap C_i|/|C'_i|$ is the *recall*, which represents the percentage of documents that are relevant to the original cluster and were in fact assigned by the algorithm. The F-score has a range of [0..1]. A larger F-score indicates that $C'_i$ and $C_i$ are more similar, and a smaller F-score indicates that they are more different.

The overall difference between the algorithm generated clustering and the original clustering is defined as the weighted average of the F-score of the component clusters: $F(\mathcal{C}', \mathcal{C}) = \sum_{i=1}^{L} \frac{|C'_i|}{|D|} F(C'_i, C_i)$, where $D = \cup_{i=1}^{L} C_i$.

There are also some other possible quality measures, such as accuracy, purity, entropy, mutual information, etc. [64, 116].

## 2.3 Document Summarization

Document summarization is still a very active research topic [119, 28], and it can be for a single document or multiple documents. A summary can be either an extract (consisting entirely of material from original input) or an abstract (at least some of the material is not present in the input) [84].

### 2.3.1 Summarization Approaches

Document summarization can be roughly categorized as the following approaches: sentences based, terms based and template based.

#### 2.3.1.1 Sentences Based Approach

In this approach, a summary consists of sentences extracted from the original documents [55, 39, 101, 85]. One advantage of this approach over others is that the summary is easy to understand for humans because it contains fluent sentences.

Some of the well-known summarizers are MEAD, WebSumm, SUMMARIST and LEAD. MEAD [100, 101] is a centroid-based extractive summarizer that scores sentences based on sentence-level and inter-sentence features which indicate the quality of the sentence as a summary sentence. It then chooses the top-ranked sentences for inclusion in the output summary. WebSumm [81] uses a graph-connectivity model and operates under the assumption that nodes which are connected to many other nodes are likely to carry salient information. SUMMARIST [55] extracts summaries based on topic signatures. LEAD is benchmark approach in which sentences are chosen from the beginning of the text.

#### 2.3.1.2 Term Based Approach

In this approach, a set of terms is used to summarize the given document(s). For document clusters, cluster descriptions (CDs) can be viewed as summarizations. Those works can be grouped as follows:

**Frequent-terms as CDs.** References [52] uses frequent terms to represent the clusters for browsing purpose, and [7, 37] use frequent term-sets to *produce* a hierarchy of clusters.

**Descriptive or Centroid-Like CDs.** In [54, 63], each cluster is described by a *descriptive* CD, consisting of a set of terms whose corresponding values in the centroid vector[1] are above a user-given threshold. Reference [41] describes a cluster by $k$ objects located near the center of the cluster.

**Discriminating CDs.** The *Cluto* clustering package [63] can generate *discriminating* CDs, which are selected from those terms that are "more prevalent in the cluster compared to the rest of the objects"(here objects mean documents).

***COBWEB* CDs.** In *COBWEB* [33], an incremental conceptual clustering algorithm, each cluster is summarized by a list of attributes, and each attribute has probability associated with.

**Others.** In [105], a document is summarized by the theme terms that obtained by use of the longer text-traversal paths from text map in chronological order. There are also other approaches to describing clusters for non-textual data. [136] uses "bounding boxes" plus some statistics to represent clusters; [43] uses multiple representatives in a cluster to represent the cluster; *CLIQUE* [1], a subspace-based clustering algorithm, generates CDs in the form of DNF expressions.

[1]The centroid vector for a collection of documents $S$ is commonly defined as $\frac{1}{|S|}\sum_{d\in S} d$, assuming that each document $d$ is represented as a TF-IDF vector.

#### 2.3.1.3 Template Based Approach

This type of summarization is based on certain pre-defined templates [89]. The filled templates can be considered as summarization. This approach involves the use of NLP and Information Extraction (IE) techniques. IE distills structured data or knowledge from un-structured text by identifying references to named entities as well as stated relationships between such entities. IE systems can be used to directly extract abstract knowledge from a text corpus, or to extract concrete data from a set of documents which can then be further analyzed with traditional data-mining techniques to discover more general patterns.

### 2.3.2 Summarization Evaluation

One major bottleneck in the development of text summarization systems is the absence of well-defined and standardized evaluation metrics [102]. Evaluating large-scale document summarization is a challenging task. Human judgment is unavoidable. Some of the commonly used evaluation metrics are [100]:

**Precision and Recall.** Precision and recall have been discussed in Section 2.2.6.

**Kappa.** Kappa is an evaluation measure which is increasingly used in NLP annotation work [100]. The Kappa coefficient $K$ controls agreement between annotators $P(A)$ by taking into account agreement by chance $P(E)$, i.e. $K = \frac{P(A)-P(E)}{1-P(E)}$. $K = 0$ when there is no agreement other than what would be expected by chance, and $K = 1$ when agreement is perfect. If two annotators agree less than expected by chance, Kappa can also be negative.

**Relative Utility.** Relative Utility (RU) [102] takes into account chance agreement as a lower bound and interjudge agreement as an upper bound of performance. RU allows judges and summarizers to pick different sentences with similar content in their summaries without penalizing them for doing so. Each judge is asked to

indicate the importance of each sentence in a cluster on a scale from 0 to 10. Judges also specify which sentences subsume or paraphrase each other. In RU, the score of an automatic summary increases with the importance of the sentences that it includes but goes down with the inclusion of redundant sentences.

**Relevance Correlation.** Relevance correlation (RC) is a new measure for assessing the relative decrease in retrieval performance when indexing summaries instead of full documents [100]. RC $r$ is defined as the linear correlation of the relevance scores ($x$ and $y$) assigned by two different IR algorithms on the same set of documents or by the same IR algorithm on different data sets:

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}}.$$

Here $\bar{x}$ and $\bar{y}$ are the means of the relevance scores for the document sequence. Based on the relevance score, one can produce a full ranking of all the summaries in the corpus.

**Content-Based Similarity Measures.** Content-based similarity measures compute the similarity between two summaries at a more finegrained level than just sentences. Two commonly used content-based similarity measures are Cosine similarity and Longest Common Subsequence [100].

**Others.** One possibility is pure human effort. Some summarizer may use several evaluation metrics together. Note that precision/recall, relative utility and kappa work only for extractive summaries [100].

For sentence based summarization approaches, there are also two properties of the summary to be measured: the Compression Ratio ($CR = \frac{lenth\ of\ summary}{length\ of\ full\ text}$) and the Retention Ratio ($RR = \frac{\#\ information\ in\ summary}{\#\ information\ in\ full\ text}$). CR measures how much shorter the summary is than the original, while RR measures how much information is retained.

## 2.4 Part-of-Speech (POS) Tagging

Part-of-speech (POS) tagging, also called grammatical tagging, is the most common form of corpus annotation. POS tagging is often seen as the first stage of a more comprehensive syntactic annotation, which assigns a phrase marker, or labeled bracketing, to each sentence of the corpus, in the manner of a phrase structure grammar [82].

The most often used POS tags is the Penn Treebank tag-set [83]. Some of the tags are listed in Table 2.1.

The task of POS-tagging assigns part of speech tags to words reflecting their syntactic category or word classes, such as noun, verb, adjective, adverb, etc. Many tagger algorithms have been developed, such as CLAWS, Xerox and the MULTEXT taggers [82].

As a side note, beyond grammatical annotations, there are also some attempts on semantic annotation. The goal of semantic annotation is try to distinguish the lexicographic senses of a given word. This is also known as "sense resolution".

## 2.5 WordNet

WordNet [87] is an online lexical reference system whose design is based on the current psycholinguistic theories of human lexical memory. WordNet was developed by the Cognitive Science Laboratory at Princeton University under the direction of Professor George A. Miller.

English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. In WordNet the lexical information is organized in terms of word meaning rather than word form. The concepts are organized into a semantic network. In the semantic model of WordNet, a word is an association between a lexicalized concept and a word form that plays a

| POS Tag | Description | Example |
|---|---|---|
| JJ | adjective | green |
| JJR | adjective, comparative | greener |
| NN | noun, singular or mass | table |
| NNS | noun plural | tables |
| NNP | proper noun, singular | John |
| NNPS | proper noun, plural | Mondays |
| RB | adverb | however, good |
| RBR | adverb, comparative | better |
| RBS | adverb, superlative | best |
| VB | verb, base form | take |
| VBD | verb, past tense | took |
| VBG | verb, gerund/present participle | taking |
| VBN | verb, past participle | taken |
| WDT | wh-determiner | which |
| WP | wh-pronoun | who, what |
| WRB | wh-abverb | where, when |

Table 2.1: Some of the Tags From Penn Treebank Tag-set

syntactic role.

WordNet has 95,600 word forms, 51,500 simple words and 44,100 collocations. It has 70,100 word meanings. The word categories are the nouns, verbs, adjectives and adverbs. WordNet has two kinds of relations, the lexical relations and the semantic relations [32].

- Lexical relations
  - Synonymy: words that have same sense. e.g. talk, speak, utter, mouth, verbalize, verbalise.
  - Antonymy: two words are antonyms if their meanings differ only in the value for a single semantic feature. e.g. dead/alive, above/below.
- Semantic relations
  - Hyponymy/Hypernymy: a word whose meaning contains the entire meaning of another. e.g. cat is a kind of animal.
  - Meronymy/Holonymy: a word is a meronym of another word (holonym) if the relation "is part of" relation holds. e.g. leg is a part of the body.
  - Entailment: a verb v1 logically entails a verb v2 when a sentence "someone v1" entails the sentence "someone v2". e.g. "snore" lexically entails "sleep".

Over the years, many people have contributed to the success of WordNet. Currently, there are many individuals applying the WordNet to their research, such as using WordNet to enhance document clustering [53, 79]. In this research, we will also use WordNet to help us for identifying opinions, which will be introduced in next chapter.

# Chapter 3

# Identifying Important Aspects of E-Rulemaking Feedback Repositories (ERFRs)

We consider stakeholders, issues and opinions as three of the most important aspects/dimensions for organizing e-rulemaking feedback repositories (ERFRs). In the following sections, we will introduce practical approaches to identify them.

## 3.1 Introduction

Following the "notice-and-comment" e-rulemaking process for a proposed rule, as described in Section 2.1, there are usually thousands and even millions of comments collected from the public. Therefore, it is highly desirable for the rule-writers and the analysts to get an overall picture about the collection before they dig into the detailed comments. For example, rule-writers might want to know the following questions as soon as they receive the feedbacks: 1) what are the important issues that the public is concerned with? 2) which groups (i.e. stakeholders) are more

concerned about these issues? 3) what are their opinions? 4) what are the typical arguments to support their opinions? etc.

Recall that one of our research goals is to produce informative summarative digest (SD) for ERFRs. We plan to organize and summarize those feedbacks based on three important factors of the feedbacks: stakeholders, issues and opinions. By considering these factors, we hope to produce an accurate and informative SD for the whole collection. In addition, by giving these three factors different priorities or weights, we can organize the feedbacks differently, thereby providing different perspectives about the collection to the users. Having said that, it is important for us to identify those three factors before further processing.

In the following sections, we will first briefly give some related works. Then we will introduce our approaches for (1) mining important issues and major stakeholders; (2) identifying opinion words related to those identified stakeholders and issues, and determining the orientation of each opinion word.

## 3.2 Related Works

It is a challenging task to automatically identify something from unstructured text. The research community has been actively working on this problem for many years using different techniques, such as NLP, information extraction (IE), statistics, association mining, classification, etc.

In this research, we need to identify stakeholders, issues and opinions from ERFR. Our work is related to but different from entity/terminology finding and sentiment classification. We briefly discuss each of them below.

### 3.2.1 Entity or Terminology Finding

There are basically two types of techniques for discovering terms in corpora: *symbolic approaches* that rely on syntactic description of terms - mainly noun phrases, and *statistical approaches* that exploit the fact that the words composing a term tend to be found close to each other and reoccurring [58, 11].

In our work, we identify the most important issues and major stakeholders from the feedbacks in a semi-automatic way. First, we use association-mining techniques to generate a list of candidates. Then, we make the final list by incorporating the input from domain experts. (Human intervention is optional.) The association mining approach can avoid producing too many non-term nouns (a shortcoming of the symbolic approaches), and the human intervention can amend the deficiency of statistical approaches where many low frequency, and some high frequency, terms are considered important.

### 3.2.2 Sentiment Classification

To automatically identify a person's opinion about something is not a trivial task. This is still a very active research topic [127, 96, 121]; previous approaches usually use classification and complicated NLP techniques. While most previous studies try to determine the sentiment at the whole-document level, we focus on the sentence level to determine the sentiment of sentences in which major stakeholders expressed opinions on some of the important issues.

References [90, 22, 56] try to classify user's opinions (e.g. thumbs-up or thumbs-downs) for certain product or product features. In [22], authors build different classifiers to classify the sentiment of whole reviews and some selected sentences. The work in [56] takes one step further than [22]: instead of classifying whole reviews, it only tries to classify the opinions on certain pre-selected product features. Their

results show that those approaches are very helpful for customer review and online opinion tracking. Our approach can achieve similar goal, but, as we will see in later chapters, we can also produce different clustering and informative summarization to meet the needs of different users.

## 3.3 Identifying Stakeholders and Issues

Before a proposed rule is published for comments, the rule-writers probably already have some ideas about the following questions: who will be the major stakeholders, and what will be the important issues of concern to the stakeholders. However, there could be some unexpected stakeholders and issues related to the proposed rule that need to be identified.

It is a challenging task to automatically identify the issues and stakeholders, and it is also hard to distinguish between issues and stakeholders. This is still an active research area as mentioned previously.

In this work, we use an alternative approach to find the issues and stakeholders. First, we use association mining to find a list of candidates containing issues and stakeholders. Then, we incorporate human judgment[1] to produce the finalized issue and stakeholder lists. This semi-automatic way works well for our needs. Note that one can also directly apply other existing techniques to get the issues and stakeholders, or can get this done by pure human efforts to bypass this component.

[1]In this dissertation, some manual checking is done to improve the result quality. One is here, which is to distinguish between issues and stakeholders. Another one is in Chapter 6, which is to adjust representative arguments (RAs) that were automatically selected for the clustering.

### 3.3.1 Association Mining for Candidates

First, we want to identify those issues that are commented in many feedbacks. We also want to identify those major stakeholders that are mentioned a lot in the feedbacks. In this work, we use the association mining [4] approach to achieve these goals, which is commonly done for this type of applications in the AI community.

Since the stakeholders and issues are usually nouns or noun phrases, we need to extract all nouns from the feedbacks first. POS tagging discussed in Section 2.4 can help us to identify the nouns. In addition, those extracted nouns need to be stored in a transaction-like format in order to apply the association mining approach.

In the following subsections, we first discuss how the documents were preprocessed, followed by some brief introduction of the association mining technique. Then we will introduce our approach to identifying stakeholders and issues.

#### 3.3.1.1 POS Tagging

POS (Part-of-speech) tagging is a very important step for us to identify the stakeholders and issues. The tagging process can parse each feedback to produce the part-of-speech tags for each word (whether the word is a noun, verb, adjective, etc). As mentioned earlier, stakeholders and issues are usually nouns or noun phrases in the feedbacks. Therefore, we should extract all the nouns and noun phrases based on POS tags.

In this dissertation, we use the Stanford POS Tagger[2] (we call it SPT) to parse each feedback and to produce the part-of-speech tags. The tags are based on the Penn Treebank tag-set, which was discussed in Section 2.4

For example, for the sentence "I strongly feel for organic crops that land fertilized with sewage sludge is wrong.", the tagged result of the SPT will be:

[2]Available at http://nlp.stanford.edu/software/tagger.shtml

I/PRP strongly/RB feel/VBP for/IN organic/JJ crops/NNS that/WDT land/VBP fertilized/VBN with/IN sewage/NN sludge/NN is/VBZ wrong/JJ ./.

From any given tagged sentence, we can easily extract all the nouns. In this research, we extract all the words that have the following noun tags: "NN", "NNS", "NNP" and "NNPS". If several adjacent words are all labeled as nouns, we consider them as noun phrase. In above example, the word "crops" and the phrase "sewage sludge" will be extracted as noun and noun phrase, respectively.

Note that the better English grammar the feedbacks have, the better results we will get from the POS tagging process.

#### 3.3.1.2 Association Mining

Association rule mining, or association mining, was first introduced in [2]. Since then it has become one of the core data-mining tasks and has attracted tremendous interest among researchers and practitioners [4, 3, 31, 73, 77, 135, 97, 76].

Association mining works as follows. Let $I$ be a set of items and $D$ a database of transactions, where each transaction has a unique identifier ($tid$) and contains a set of items called an *itemset*. An itemset with $k$ items is called a *k-itemset*. The *support* of an itemset $X$, denoted $sup(X)$, is the number of transactions in which that itemset occurs as a subset. A *k-subset* is a $k$-length subset of an itemset. An itemset is frequent or large if its support is more than a user-specified *minimum support (minsup)* value. A frequent itemset is maximal if it is not a subset of any other frequent itemset.

An *association rule* is an expression of the form $A \Rightarrow B$, where $A$ and $B$ are itemsets. The rule's support is the joint probability of a transaction containing both $A$ and $B$, and is given as $sup(A \cup B)$. The *confidence* of the rule is the conditional probability that a transaction contains $B$, given that it contains $A$

and is given as $\frac{sup(A \cup B)}{sup(A)}$. A rule is *frequent* if its support is greater than *minsup*, and *strong* if its confidence is more than a user-specified *minimum confidence (minconf)*.

There are usually two steps to generate all association rules that have a support greater than *minsup* (the rules are frequent) and confidence greater than *minconf* (the rules are strong).

1. Find all frequent itemsets having minimum support. The search space for enumeration of all frequent itemsets is $2^m$, which is exponential in the number of items $m$.

2. Generate strong rules having minimum confidence from the frequent itemsets. Rules with the form of $X \setminus Y \Rightarrow Y$, where $Y \subset X$ and $X$ is frequent, will be generated and tested against the *minconf* threshold. Because each subset of $X$ as the consequent has to be considered, the rule-generation step's complexity is $O(p \cdot 2^q)$, where $p$ is the number of frequent itemsets, and $q$ is the longest frequent itemset. ■

#### 3.3.1.3 The Proposed Approach

We now turn to our approach to identify stakeholders and issues. The overall process is the following:

- POS tagging. We use the SPT to parse each feedback and to produce the part-of-speech tags.

- Extracting nouns. We extract nouns or noun phrases from each tagged feedback to create a transaction file; in that file, each line contains words from one feedback. The tags considered as nouns are "NN", "NNS", "NNP" and "NNPS". We save the nouns or noun phrases to a file so that it can be used in the pruning step.

- Pre-processing. Pre-processing of words is also performed, which includes removal of stopwords and stemming as discussed in Section 2.2.2.
- Association mining. We run the Apriori association-mining algorithm on the transaction file. We only need to obtain frequent itemsets, and do not need to generate the rules.
- Pruning and compacting. We also perform some redundancy and compactness pruning on the final frequent sets. ■

The frequent itemsets will be considered as candidates of stakeholders and issues. In this work, we only consider frequent 1-itemsets, for simplicity.

### 3.3.2 Incorporating Human Input

The candidates generated by the association mining will probably cover most of the issues and stakeholders. However, not all candidates, especially some of the most frequent ones, are important issues and major stakeholders. Furthermore, because the candidates contain both issues and stakeholders, it is hard to distinguish them without human intervention.

In this work, we employ human effort to go though the list and let the human to identify which ones are issues and which are stakeholders. Users can modify the lists based on their knowledge about the proposed rules and the feedbacks.

Note that we treat pronouns (I, we, you, etc.) as stakeholders to help identifying opinions, which will be discussed later. Also, in our experiments, we set the number of issues and the number of stakeholders to around 20, for simplicity.

### 3.3.3 Examples

To illustrate the idea, let us look at some actual feedbacks from the dataset *DoA-NOP* as described in Section 2.1.4. Note that some of the feedbacks contain spelling

| Ref. | Original Feedback Sentences |
|---|---|
| A.1.1 | I strongly oppose the proposed rules for organics as currently written. As a business person, an educator, and a cancer patient, I find the efforts of USDA laudable but seriously lacking in several key areas. ... |
| A.1.2 | I am outraged as a consumer that irradiated foods, gentically engineered foods, and foods grown on lands fertilized with sewage sludge are included in the USDA's proposed rules for organic foods. ... |
| A.1.3 | I am a consumer of organic food products and wish that methods such as sewage sludge, irradiation, and biotechnology be banned from the production of organic foods. ... |

Table 3.1: Portions of Three Feedbacks From the DoA-NOP Dataset

and grammar errors, and we just use them as they are.

Table 3.1 lists portions of three feedbacks from the the DoA-NOP Dataset. The complete original feedbacks are listed in Appendix A; the "Ref." column of the tables contains the identifiers of the feedbacks as listed in the appendix.

By using the SPT on those selected sentences, we get the tagged sentences shown in Table 3.2. We extract all the nouns and noun phases from those tagged results. For clarity, the nouns or noun phrases are separated by comma and the results are shown in Table 3.3. As mentioned earlier, we save all the nouns and nouns phrases to a transaction file. For those nouns or noun phrases that appeared multiple times[3] in a feedback, we only keep one in the file.

Next, we convert those extracted nouns and noun phrases to lower cases (e.g. USDA to usda). We then apply some preprocessing procedures, such as stopwords removal and stemming. In this example, stopwords removal does not remove any word. The stemming process will change the "rules" to "rule", "foods" to "food", etc.

[3]Case insensitive comparison

| Ref. | Sentences With POS Tags |
|---|---|
| A.1.1 | I/PRP strongly/RB oppose/VBP the/DT proposed/VBN rules/NNS for/IN organics/NNS as/IN currently/RB written/VBN ./. As/IN a/DT business/NN person/NN ,/, an/DT educator/NN ,/, and/CC a/DT cancer/NN patient/NN ,/, I/PRP find/VBP the/DT efforts/NNS of/IN USDA/NNP laudable/JJ but/CC seriously/RB lacking/VBG in/IN several/JJ key/JJ areas/NNS ./. ... |
| A.1.2 | I/PRP am/VBP outraged/JJ as/IN a/DT consumer/NN that/WDT irradiated/VBD foods/NNS ,/, gentically/RB engineered/VBN foods/NNS ,/, and/CC foods/NNS grown/VBN on/IN lands/NNS fertilized/VBN with/IN sewage/NN sludge/NN are/VBP included/VBN in/IN the/DT USDA's/NNP proposed/VBD rules/NNS for/IN organic/JJ foods/NNS ./. ... |
| A.1.3 | I/PRP am/VBP a/DT consumer/NN of/IN organic/JJ food/NN products/NNS and/CC wish/VBP that/IN methods/NNS such/JJ as/IN sewage/NN sludge/NN ,/, irradiation/NN ,/, and/CC biotechnology/NN be/VB banned/VBN from/IN the/DT production/NN of/IN organic/JJ foods/NNS ./. ... |

Table 3.2: Feedbacks in Table 3.1 With POS Tags

| Ref. | Nouns or Noun Phrases |
|---|---|
| A.1.1 | rules, organics, business person, educator, cancer patient, efforts, USDA, areas ... |
| A.1.2 | consumer, foods, lands, sewage sludge, USDA's, rules ... |
| A.1.3 | consumer, food products, methods, sewage sludge, irradiation, biotechnology, production, foods |

Table 3.3: Nouns or Noun Phrases Extracted From the Feedbacks

| Terms | Support % | Origin Feedbacks |
|---|---|---|
| rule | 67 | A.1.1, A.1.2 |
| usda | 67 | A.1.1, A.1.2 |
| food | 67 | A.1.2, A.1.3 |
| consumer | 67 | A.1.2, A.1.3 |
| sewage sludge | 67 | A.1.2, A.1.3 |

Table 3.4: Frequent Terms Mined From the Nouns and Noun Phrases

| Terms | Is Stakeholder | Is Issue |
|---|---|---|
| consumer | Yes | |
| usda | Yes | |
| rule | | Yes |
| sewage sludge | | Yes |

Table 3.5: User Identified Stakeholders and Issues

After the preprocessing, we apply the association mining algorithm to mine the frequent itemsets. In this example, we use 67% as minimum support *(minsup)*, i.e terms appeared in at least two feedbacks.

From Table 3.4 we can see that those frequent terms can be either stakeholders or issues. This list, as candidates, will be provided to human users. Human users will be the ultimate judge to distinguish between stakeholders and issues, and they can also add or delete terms from the list.

Hypothetically, after users review the list, they decide to select two stakeholders and two issues from the above examples. Also, no additions or deletions were made. The result is illustrated in Table 3.5.

Note that there is no need for the pruning and compacting process in this example. However, if there were too many frequent terms, the terms with lower

support value will be pruned. In addition, suppose both "food", and "food product" are identified as frequent items, the compacting process will remove the term "food", since we prefer phrase over single term.

In Chapter 5, we will see that the results of our approach on some large e-rulemaking data sets are also very good. Those generated candidates are very comprehensive as for identifying stakeholders and issues. Some of the results can also be seen in Appendix B.1.

## 3.4 Identifying Opinions

Opinion words are used to express subjective opinions. For the data we are dealing with, e-rulemaking feedbacks, they are full of opinions. The person who submitted those comments may have expressed opinions on some unimportant issues, or they may have expressed opinions on some other issues that are not related to the proposed rule at all.

In this work, we only focus on those opinions that were (i) expressed by the identified stakeholders (ii) and/or have been expressed on those identified issues. If a sentence contains such opinion expression, we call it *argument*, which will be discussed in more detail in Chapter 6. Note that multiple stakeholders could express different opinions on one or more issues.

We now discuss our approaches to extract opinion words and determine opinion word orientation.

As mentioned in Section 3.2.2, it is a challenging task to identify opinions and their orientation [127, 96, 121]. In this work, we identify the opinion terms by using shallow syntax parsing on the POS tagging results, and also utilizing some *cue phrases*. For determining the orientation, we check the orientation of a given term against a list of (semi-automatically) selected terms with known orientations.

Below, we will introduce each of those techniques in turn.

### 3.4.1 Extracting Opinion Words

Existing studies show that adjectives are usually used for expressing subjective opinions [127, 96]. In this work, we will mainly focus on adjectives (tagged as "JJ" or "JJR") as opinion words; but, we also consider some verbs provided in the cue phrases. For example, *bad* is an opinion word in "This rule is bad for the environment", and *oppose* is also an opinion word in "I strongly oppose this rule".

We extract opinion words using the following procedure:

---

**Procedure 3.4.1** Extracting Opinion Words

*for each sentence in the feedback* {
    *if (it contains issues or stakeholders)* {
        *extract all the adjectives as opinion words;*
        *record this sentence as an argument;*
    } *else if (it contains cue phrases)* {
        *extract the cue word as opinion word;*
        *record this sentence as an argument;*
    } *//endif*
} *//endfor*

---

We use about 10 cue phrases in above procedure, including: *? oppose, ? disagree, ? support, ? appraise, ? suggest, ? hope*, etc. The ? represents personal pronouns (i.e. I, we, etc.). It should be noted that cue phrases may indicate a positive opinion (such as *support* and *appraise*), a negative opinion (such as *oppose* and *disagree*) or others (such as *suggest* and *hope*).

It is worth to mention that we can specify the maximum word distance between the pronoun and the cue word. We choose the maximum distance to be 2 in this

work. This allows us to handle the cases when adverbs are used. For example, "I oppose ...", "I strongly oppose ..." and "I also strongly oppose ..." will all be considered as argument, since the distance between "I" and "oppose" are 0, 1 and 2, respectively. However, such setting is unable to recognize the following format as an argument: "I, as a teacher, oppose ...".

### 3.4.2 Determining Orientation of Opinion Words

We now turn to determine the semantic orientation of each opinion word identified in the previous step. The orientation will be used to predict the semantic orientation of each argument. Note that some opinion words may have positive orientation (i.e. in favor of) in their semantic group, some have negative orientation (i.e. against) and others may not have orientation (i.e. new suggestions or don't know/care). In this work, we classify the orientation into three categories: *positive*, *negative* and *unknown*.

In this work, we use WordNet, as discussed in Section 2.5, to help us in orientation determination. WordNet uses synset (synonym set) to organize words with similar meaning (sense). In other words, the synset of a word contains synonyms of that word. Therefore, we can tell the orientation of a given word, if we know the orientation about one of their synonyms or antonyms.

Our approach to determine orientation is the following:

---

**Procedure 3.4.2** Determining Opinion Words Orientation

1. *Manually select a list of seed words with known orientation with the format of* <word, type, orientation>*, where* type *can be* *adjective or verb; and* orientation *can be* *positive or negative;*
2. *Repeatedly grow the seed list by adding synonyms and antonyms of adjectives using the WordNet until the list is large enough;*

*3. Determine the orientation of a given opinion word based on the above list;*

---

For step 1, we pick a small number of words as initial seeds (around 30); some of these words have positive orientation, such as <*support, v, p*>, <*beneficial, adj, p*>, <*useful, adj, p*>, etc., and some have negative orientation, such as <*oppose, v, n*>, <*awful, adj, n*>, <*unfair, adj, n*>, etc.

In step 2, we use all adjectives to expand the seed list by searching their synsets in the WordNet. (We do not expand the verbs.) For each adjective, we add their synonyms and antonyms to the seed list if they haven't been added yet. The orientation of their synonyms will be the same as the given word, and that of their antonyms will be the opposite of the original word. Note that some words may have multiple senses; in this work we only use the top sense to grow the list. This process can be repeated a number of times, until the seed list is large enough or the change is limited with additional run. In this work, our Java implementation utilizes the JWNL[4] package, and the pseudo code is shown in Procedure 3.4.3.

---

**Procedure 3.4.3** Expanding the List of Opinion Words

```
for each word w_i in the seed list {
    get all senses of w_i by looking up the dictionary;
    if (there is more than 1 sense AND w_i is an adjective) {
        call getSynonyms(sense[0]) to get synonyms;
            set the orientation of synonyms same as the one w_i has;
        call getAntonyms(sense[0]) to get antonyms;
            set the orientation of antonyms opposite to the one w_i has;
    } //endif
} //endfor
```

[4]JWNL is an open source Java API for accessing WordNet-style relational dictionaries (WordNet 2.0 compatible). http://jwordnet.sourceforge.net/

We can see that if the word list is large enough we can almost predict the orientation of all opinion words. Thus, in step 3 if a given opinion word can be found in the candidate pool, its orientation will be set accordingly (*positive* or *negative*); otherwise it will be set as *unknown*. It is worth mentioning that the orientation of the seed words was assigned based on the top sense in WordNet. The orientation may not be true for some context. Therefore, human eyeballing of the generated list is recommended to improve the accuracy.

Note that if there is a negation word such as "no" or "not" around the opinion word, we treat the orientation to be opposite to the orientation of the opinion word self. For example, "we do not support ..." will be considered as *negative*. In this work, we let users to specify the maximum distance between the negation word and the opinion word. We choose that value to be 2 in our experiments.

### 3.4.3 Examples

In Chapters 5 and 6, we will see more results for some large data sets. Below, we will illustrate the idea of Procedure 3.4.2 by using the examples listed in Table 3.1. Recall that we only focus on those opinion words that are associated with some identified stakeholders and/or issues.

There are two example sentences in feedback A.1.1. The first sentence contains one of the identified issue "rule", and the second contains the stakeholder "usda". By parsing those two sentences, we can see that the first sentence contains the opinion word "oppose", which is one of the predefined cue words that has negative orientation. The second sentence contains three adjective words (tagged as "JJ"): "laudable", "several" and "key". By looking up the seeds list, we find that "laudable" has positive orientation and other two words have no orientation. However, because of the word "but", the overall orientation of the second sentence

is negative[5]. In addition, both sentence will be considered as arguments.

The example sentence in A.1.2 contains both stakeholders and issues. There are words tagged as adjectives: "outraged", which has negative orientation; and "organic", which has no orientation.

Similarly, the example sentence in A.1.3 has in it both stakeholder and issue. There are two tagged as adjectives: "such" and"organic". Both of them have no orientation.

## 3.5 Summary

For large e-rulemaking feedback repositories (ERFRs), we considered stakeholders, issues and opinions as three of the most important aspects/dimensions for organizing them. In this chapter, we have introduced our approaches to identify stakeholders, issues and opinions.

We first preprocessed those feedbacks to get the Part-Of-Speech (POS) tags. Based on the POS tags, we extracted nouns and noun phrases to generate transaction files. We then apply the association mining algorithm on those transaction files to obtain a list of candidates that can be considered as stakeholders and issues. Those candidates will be provided to human users, and the human users will make the ultimate judgment about what are issues and what are stakeholders.

It is a very challenging task to automatically identify a person's opinion about something. In this chapter, we introduced a practical, yet very effective, approach to identify opinions. We only focused on those opinions that are expressed by some identified stakeholders or/and have been expressed on some identified issues.

We only considered adjectives and limited number of verbs as opinion words. To be an opinion word, there should be stakeholders or/and issues associated with

[5]Interestingly, there are more negative opinions than positive ones in the feedbacks.

them in the same sentence. We identified all the adjectives by parsing the tagged sentences, and verbs by looking up the predefined cue phrases. The orientation of the opinion words was determined by utilizing the synset (synonym set) organization of WordNet, since WordNet uses synset to organize words with similar meaning. We first provided a small list of words with known orientation. Then, we repeatedly grown the list by looking up the WordNet. For any given word, the orientation was determined by looking up the list.

In this chapter, we have also used some example feedbacks to illustrate the approaches. In later chapters we will see how those stakeholders, issues and opinions are used to produce SDs.

# Chapter 4

# Constructing Cluster Descriptions for Document Clustering

We believe that good cluster descriptions (CDs) are important components of good document clusterings, and are crucial for managing large document repositories, such as e-rulemaking feedback repositories (ERFRs).

In this chapter, we will study the problem of CDs for any given document clustering regardless of the clustering algorithm used to produce the clusters. First, we will give our definition on CDs. Then we will discuss and formalize how to interpret the CDs and how to resolve perception competition of CDs. We also present a novel *CD-based classification* approach to systematically evaluate CD quality. After introducing and examining some surrogate quality measures for efficiently constructing CDs, we will give several effective search algorithms, namely PagodaCD and *CumulativeCD*, for constructing CDs. Then we will show some experimental results on constructing CDs by utilizing several subsets of the Reuters documents collection. At the end of this chapter, we will present our initial efforts on using genetic algorithm(GA) to construct CD.

## 4.1 Introduction

Large document repositories need to be organized, summarized and labeled in order to be used effectively. Cluster labels are essential for users to efficiently get a high-level sense of what the clusters contain, and for use as conceptual "handles" to the clusters. Without such labels, users will need to browse many documents in the clusters to get that sense. Human labeling of clusters is not viable when clustering is performed on demand or for few users. It is desirable to automatically generate cluster labels, or succinct and informative cluster descriptions (CDs), so that users can get that sense about the clusters by just examining the CDs. Such CDs can also be used as hints for producing final cluster labels by humans.

Much research has been done on document clustering. However, previous clustering algorithms mainly focused on cluster formation, and paid little attention to producing CDs. Even when CDs were generated [7, 37, 52, 63, 33], they were often just by-products of the clustering process: [7, 37, 52] use the most frequent terms as CDs, [41, 54, 63] use "descriptive" or centroid-like terms as CDs, [63] use "discriminating" terms as CDs, and [33] use terms and their frequency distributions as CDs. Except [33], these approaches did not treat CDs as primary product to generate. Furthermore, none of them addressed the diversity factor on the terms in CDs, and the quality of CDs has not been thoroughly addressed, to the best of our knowledge. While there are approaches that produce a short summary for multiple documents by extracting some key phrases or sentences [55, 28, 84], our study is focused on succinct and informative CDs consisting of a set of terms. We believe that such CDs is more useful for cluster labeling.

We propose a CD-based classification for simulating how to interpret CDs; the corresponding classifier only uses the CDs and their associated interpretation in making classification decisions. We then propose to use the F-score of the classification to measure CD quality. This classification approach also allows us to

resolve cluster competition in the interpretation process. Competition occurs when competing evidence exists regarding to which cluster a given document should be assigned.

Using F-score directly to search for high quality CDs is too expensive. We need some "surrogate" measures of F-score for efficient search. In this work we consider the CDD measure which combines the three factors of *coverage*, *disjointness* between terms across CDs for different clusters, and *diversity* among terms within the CD of one cluster. Notice that diversity measures overlap among terms in the CD of one cluster, whereas disjointness measures overlap among terms in CDs of different clusters. We will argue that diversity is important in capturing the different flavors of a given cluster. Diversity has not been considered explicitly in previous work on CD construction.

We give a search algorithm, namely PagodaCD, for constructing CDs. PagodaCD is a layered improvement-based replacement algorithm, and it uses the CDD surrogate quality measure. We also preselect a set of candidate terms to reduce computation cost. Experimental evaluation on subsets of the Reuters collection shows that the PagodaCD algorithm is efficient, and it can produce high quality CDs. CDs produced by PagodaCD also has the monotone quality behavior, giving higher quality CDs when more terms are in the CDs.

## 4.2 Related Works

Roughly speaking, in this work we study CD in the form of small term sets for any given document clusters, and address the issues of how to measure the quality of CDs and how to construct high quality CDs. Related works, as briefly mentioned in Section 2.3.1.2, can be categorized as follows:

**Frequent-terms as CDs.** Reference [52] uses frequent terms to represent clusters

for browsing. References [7, 37] use frequent term-sets to *produce* a hierarchy of clusters and those frequent terms can be considered as CDs.

**Descriptive or Centroid-like CDs.** In [54, 63], each cluster is described by a *descriptive* CD, consisting of a set of terms whose corresponding values in the centroid vector[1] are above a user-given threshold. Reference [41] describes a cluster by $k$ objects located near the center of the cluster.

**Discriminating CDs.** The *Cluto* clustering toolkit [63] also generates *discriminating* CDs, which are selected from those terms that are "more prevalent in the cluster compared to the rest of the objects"(here objects mean documents).

***COBWEB* CDs.** In *COBWEB* [33], a conceptual clustering algorithm, each cluster is summarized by a list of attributes and associated probabilities.

Notice that these term-based approaches did not address the diversity factor on the terms in CDs. The quality of CDs as cluster labels has not been thoroughly addressed, to the best of our knowledge.

**Others.** There are approaches that try to produce a short summary for multiple documents by extracting some key phrases or sentences [55, 28, 84]. In contrasts, our study is focused on succinct and informative CDs consisting of a set of terms. Some of the other approaches extract information from documents based on certain pre-defined templates [89]. The filled templates can be considered as some kind of CDs. This approach involves the use of NLP and Information Extraction (IE) techniques, which is different from our term-based approach.

There are also other approaches to describing clusters for non-textual data. [136] uses "bounding boxes" plus some statistics to represent clusters; [43] uses multiple representatives in a cluster to represent the cluster; *CLIQUE* [1] generates CDs in the form of DNF expressions.

[1]The centroid vector for a collection of documents $S$ is commonly defined as $\frac{1}{|S|}\sum_{d\in S} d$, assuming that each document $d$ is represented as a TF-IDF vector.

## 4.3 Cluster Description (CD)

Let $D$ be a given collection of documents. A document is a set of terms (namely words or phrases). A *clustering*[2] $\mathcal{C}$ consists of a number $L$ of clusters, $C_1, C_2, ..., C_L$, of all the documents in $D$.

Roughly speaking, a cluster description is intended to be used as a succinct cluster label. Formally, we have:

**Definition 4.3.1** A *cluster description* (CD) for a cluster $C$ is a set of $k$ terms. A *clustering description* for a clustering $\mathcal{C}$ consists of $L$ cluster descriptions $CD_1, ..., CD_L$, one for each cluster $C_i$. ■

To allow easy interpretation, $k$ should be a fairly small number, such as between 1 and 20. Constraints can be imposed on the terms in a CD. For example, we can require a CD to contain only terms that occur in some documents in its cluster. While we consider document clusters only here, one can also consider CDs for non-document clusters.

We will address the following issues, which have not been considered by previous studies to the best of our knowledge, in the rest of this Chapter: (i) how to interpret CDs, (ii) how to measure the quality of CDs, and (iii) how to produce high-quality CDs.

## 4.4 CD Interpretation and Quality

To be useful as descriptive "labels" to clusters, CDs should allow users to get a rough picture of the contents of the clusters; they should get such a picture by looking at the CDs (but not the actual contents of the clusters) and mentally interpreting them in some natural manner. The interpretation can be viewed as

[2]Clustering is used as a noun here.

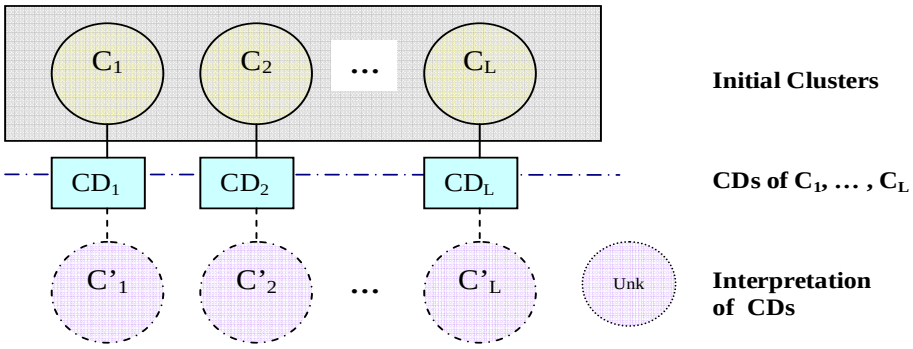



Figure 4.1: Clusters, CDs and Interpreted Clusters

a mapping from CDs to the interpreted clusters; the interpreted clusters contain what users believe are in the clusters. The amount of difference between the interpreted and the original clusters can then be used to measure the quality of the CDs. We formalize the interpretation process and consider how to measure the difference below[3].

### 4.4.1 Interpretation via CD-Based Classification

Suppose the original clusters are $C_1, ..., C_L$, and their corresponding CDs are $CD_1, ..., CD_L$. The interpretation can be illustrated in Figure 4.1. The initial clusters are only provided to show the entire picture; users do not need to examine them during interpretation.

CD interpretation can be formalized in different ways. We believe that a natural way is the following: a user combines his/her understanding or interpretation of the individual terms in the CDs to form a rough picture of the clusters' contents. We capture user understanding or interpretation of individual terms as follows.

[3]To our best knowledge, we are the first to consider interpreting CDs and use that interpretation to measure CD quality.

**Definition 4.4.1** The *interpretation* of a term $t$ w.r.t. an underlying universe $S$ of documents, denoted as $\mathsf{INT}_S(t)$, is the set of documents in $S$ containing the term $t$: $\mathsf{INT}_S(t) = \{d \mid d \in S \text{ such that } t \in d\}$. We will omit $S$ when $S$ is the collection $D$ of all documents under consideration. ■

While $\mathsf{INT}_S(t)$ is semantically the same as the concepts of tid-set, cover or SAT previously used in the literature, we use the notation of $\mathsf{INT}$ to emphasize that these sets are the basis of users' perception of the terms. Notice that one can also consider other factors such as synonyms of terms when defining $\mathsf{INT}_S(t)$ by including a document $d$ in $\mathsf{INT}_S(t)$ if $d$ contains a synonym of $t$).

When interpreting CDs, users form virtual or interpreted clusters by assigning documents to clusters based on their intuition and some "rough mental reckoning". Since a term $t$ can occur in different clusters, there is competition in the interpretation of $t$ with respect to the "right" cluster. On the other hand, since a document $d$ can contain terms from CDs of multiple clusters, there can be competition regarding which cluster to assign $d$ to: if $d$ contains a term $t_1 \in CD_1$ and a term $t_2 \in CD_2$, then competition occurs since $t_1$ indicates that $d$ should belong to $C_1$ and $t_2$ indicates that $d$ should belong to $C_2$.

We combine the interpretation of the terms in CDs and resolve the competition to form interpretations for all clusters by using the *CD-based classification* approach. Here, we use the terms in the CDs as a classifier to classify documents into interpreted clusters as follows:

---

**Algorithm 4.4.1** The CD-based classification

*1. For each document $d$ and cluster $C_i$, define*[4]

$$\mathsf{Score}(d, C_i) = \frac{|\cup_{t \in d \cap CD_i} \mathsf{INT}_{C_i}(t)|}{|C_i|}.$$

[4]We use $|S|$ to denote the cardinality of a set $S$.

*Each document $d'$ in $\mathsf{INT}_{C_i}(t)$ gives a signal regarding the membership of $d$ in $C_i$, where $t \in d \cap CD_i$. By using the union of INTs, this score uses the signal contained in any given document $d'$ exactly once.*

*2. A document $d$ is assigned to the interpreted cluster $C'_i$ if $d$ has the highest score at cluster $C_i$ (i.e., $\mathsf{Score}(d, C_i) = max\{\mathsf{Score}(d, C_j) \mid 1 \leq j \leq L\}$). If the highest score is zero, then $d$ is assigned to the unknown cluster. We break ties by assigning $d$ to the first cluster (in some fixed order) having the highest score.* ■

---

Collectively, the interpreted clusters $C'_1, ..., C'_L$ will be referred to as the *interpretation* of the CDs using the *CD-based classification* approach. While Score combines terms using roughly the OR, other logical connectives can also be used.

| $CD_1 = \{a, c\}$ | $CD_2 = \{g, i\}$ |
|---|---|
| $d_{11}$: a b c d | $d_{21}$: e g h i |
| $d_{12}$: a c e f | $d_{22}$: c i j |

(a) Two given clusters and their CDs

| $C'_1$ | $C'_2$ |
|---|---|
| $d_{11}$: a b c d | $d_{21}$: e g h i |
| $d_{12}$: a c e f | |
| $d_{22}$: c i j | |

(b) The interpreted clusters

Table 4.1: The Interpreted Clusters Using CD-based Classification

**Example 4.4.1** *We now use the example given in Table 4.1 to illustrate. Suppose we are given two clusters $C_1 = \{d_{11}, d_{12}\}$ and $C_2 = \{d_{21}, d_{22}\}$, and two CDs $CD_1 = \{a, c\}$ and $CD_2 = \{g, i\}$ (See (a)). To evaluate the quality of the given CDs, we apply our* CD-based classification *approach to those documents. The interpreted clusters formed by this process are shown in (b). Consider document $d_{22}$. It contains $c$ in $CD_1$ and $i$ in $CD_2$. Both $d_{11}$ and $d_{12}$ of $C_1$ contain $c$, so $score(d_{22}, C_1) = |\{d_{11}, d_{12}\}|/2 = 1$. Only $d_{22}$ in $C_2$ contains $c$ or $i$, so $score(d_{22}, C_2) = |\{d_{22}\}|/2 = 0.5$. Since $score(d_{22}, C_1) > score(d_{22}, C_2)$, we assign it to $C'_1$. Note that the scores are calculated based on the contents of the original clustering.* ■

### 4.4.2 F-score as Measure of Quality

We measure the quality of CDs by using the amount of difference between the original and the interpreted clustering. We measure the difference using F-score [123], also called *F-measure*, a widely used quality measure for clustering and information retrieval methods.

Suppose the original clustering is $\mathcal{K} = \{C_1, ..., C_L\}$, the corresponding CDs are $CD_1, ..., CD_L$, and the interpreted clustering of the CDs is $\mathcal{K}' = \{C'_1, ..., C'_L\}$. For each $i$, the F-score for $C'_i$ and $C_i$, denoted by $F(C'_i, C_i)$, is defined as $F(C'_i, C_i) = \frac{2*P(C'_i,C_i)*R(C'_i,C_i)}{P(C'_i,C_i)+R(C'_i,C_i)}$, where $P(C'_i, C_i) = |C'_i \cap C_i|/|C_i|$ is the *precision*, $R(C'_i, C_i) = |C'_i \cap C_i|/|C'_i|$ is the *recall*. The F-score has a range of [0..1]. A larger F-score indicates that $C'_i$ and $C_i$ are more similar, and a smaller F-score indicates that they are more different.

The overall difference between the interpreted clustering and the original clustering is defined as the weighted average of the F-score of the component clusters: $F(\mathcal{K}', \mathcal{K}) = \sum_{i=1}^{L} \frac{|C'_i|}{|D|} F(C'_i, C_i)$, where $D = \cup_{i=1}^{L} C_i$. We use $F(\mathcal{K}', \mathcal{K})$ as our measure of CD quality.

## 4.5 Surrogate CD Quality Measures for Efficient Search

Using F-score to directly search for good CDs is too expensive. So we need to give efficient surrogate quality measures for use in the search process. In this section, we introduce one such measure, namely the CDD measure, which combines the three factors of *coverage*, *disjointness*, and *diversity*.

Intuitively, *coverage* is used to encourage the selection of terms with high frequency (matching large number of documents) in a given cluster, *disjointness* is

used to discourage the selection of terms with high inter-cluster overlap, and *diversity* is used to discourage the selection of terms with high intra-cluster overlap. Consequently, the three factors help us to capture the quality measure discussed in Section 4.4.

We first show that using F-score to directly search for good CDs is expensive. Let $L$ denote the number of clusters, $\gamma$ the number of candidate terms for constructing CD in a cluster, $k$ the desired size (number of terms) of the CD for a cluster, $|D|$ the total number of documents, and $\tau$ the number of unique terms. Given the CDs for a clustering, using F-score directly to check and pick the best term for one single-term replacement will require at least $O(L|D|\tau/32 * Lk\gamma) = O(L^2\gamma k|D|\tau/32)$ operations. (The $L|D|\tau/32$ term is the minimum cost of computing the F-score of the clustering CDs, involving at least finding the terms in the given CDs contained in each document $d$ (and a number of union operations on bit sets, which is ignored in the formula). The $Lk\gamma$ term represents the number of potential replacements of the terms in the old CDs.) In our experiments, the averages of the values of these parameters are: $L = 10$, $\gamma = 60$, $k = 10$, $|D| = 5000$, and $\tau = 17500$. Assuming those values, $O(L^2\gamma k|D|\tau/32) \approx 1.64 * 10^{11}$. In contrast, using our CDD surrogate measure, we need just $9.37 * 10^5$ operations, only about $5.7 * 10^{-6}$ of the cost of the direct F-score based search. Since we need to repeat this single-term replacement many times, we cannot afford to use F-score to search directly.

### 4.5.1 Three factors

We now discuss the three factors of *coverage*, *disjointness*, and *diversity*, which we will use as surrogates for F-score in the search process. While the *disjointness* is defined on CDs for one clustering, the other two are on CDs for one cluster.

To describe the contents of the clusters well, a CD must represent or cover the cluster well: A good CD for a cluster $C$ is a term set $T$ where $\mathsf{Cov}_C(T)$ is large.

**Definition 4.5.1** *The* coverage *of a $CD = T$ for a cluster $C$ measures how well a term set $T$ covers $C$, and is defined by*

$$\mathsf{Cov}_C(T) = \frac{|\bigcup_{t \in T} \mathsf{INT}_C(t)|}{|C|}.$$

■

To avoid the adverse impact of competition, the CDs for different clusters should have minimal competition against each other: good CDs for a clustering $C_1, ..., C_L$ is a set of CDs such that $\mathsf{Dis}(CD_1, ..., CD_L)$ is large.

**Definition 4.5.2** *Let $CD_1, ..., CD_L$ be a CD for a given clustering $C_1, ..., C_L$.* Disjointness *measures overlap between terms in different CDs, and is defined by*

$$\mathsf{Dis}(CD_1, ..., CD_L) = \frac{1}{\sum_{1 \le i,j \le L,\, i \ne j} |\mathsf{INT}_{C_j}(CD_i)| + 1}.$$

■

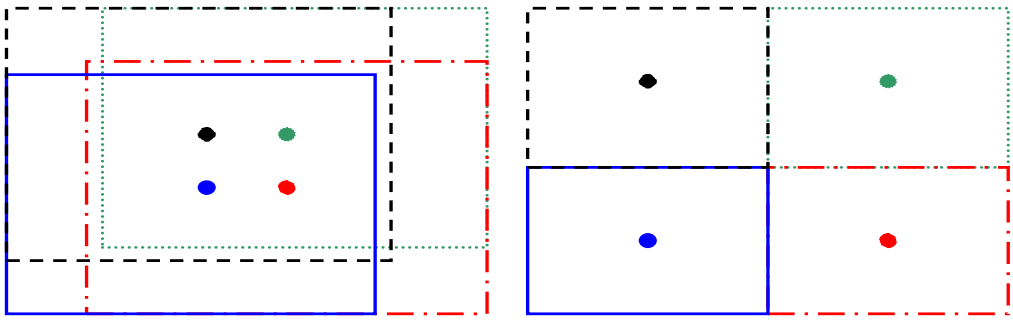
The terms in a good CD should be as different as possible (less overlap among INTs): A good CD for a cluster $C$ is a term set $T$ such that $\mathsf{Div}_C(T)$ is large.

**Definition 4.5.3** *The* diversity *of a $CD = T$ for a cluster $C$ measures overlap among terms within $T$, and is defined by*

$$\mathsf{Div}_C(T) = \frac{1}{\sum_{t,t' \in T,\ t \ne t'} |\mathsf{INT}_C(t) \cap \mathsf{INT}_C(t')| + 1}.$$

■

To see why $\mathsf{Div}_C(T)$ is important, consider the cluster $C$ depicted in Figure 4.2. Suppose $T = \{t_1, t_2, t_3, t_4\}$ and $T' = \{t'_1, t'_2, t'_3, t'_4\}$ are two candidate term sets. Suppose the unions of their INTs are the same, i.e. $\cup_{i=1}^{4} \mathsf{INT}_C(t_i) = \cup_{i=1}^{4} \mathsf{INT}_C(t'_i)$. Suppose further that overlap in (Figure 4.2.a) is much larger than overlap in (Figure 4.2.b). Metaphorically speaking, a term $t$ can be viewed as the centroid of



(a) $T'$: Less Diverse CD (b) $T$: More Diverse CD

Figure 4.2: Importance of Diversity

$\mathsf{INT}_C(t)$. The centroids are much closer to each other in Figure 4.2.a than in Figure 4.2.b. As a consequence, it is much harder to synthesize the whole picture of the entire cluster using $T'$ than using $T$.

In general, when the centroids are close to each other, it is hard to synthesize the whole picture of the entire cluster; in contrast, when they are more widely and evenly distributed, they can be combined to offer better picture of the whole cluster. The importance of diversity can also be seen from an analogy: diversity is important [21] for the performance of classifier ensembles [107, 12], and the terms in a CD play a similar role for the collective interpretation of the CD as the committee-member classifiers in the collective classification.

### 4.5.2 The CDD Measure

We now define the CDD surrogate measure in terms of the three factors. For use in the search process, we are interested in comparing two CDs, a new and an old, to determine the quality improvement offered by the new over the old. We will first define improvement for the factors, and then combine them to form improvement of the CDD measure.

Let $C_1, ..., C_L$ be a given clustering. Let $CD^o_1, ..., CD^o_L$ and $CD^n_1, ..., CD^n_L$ be

two (an old and a new) CDs for the clustering. We require that the new be obtained from the old by modifying[5] just one of the $CD_i^o$'s, keeping the others unchanged; let $CD_j^o$ be the $CD_i^o$ that is modified.

The *improvement* of the factors are defined as:

$$\delta(\mathsf{Cov}) = \frac{\mathsf{Cov}_{C_j}(CD_j^n) - \mathsf{Cov}_{C_j}(CD_j^o)}{\mathsf{Cov}_{C_j}(CD_j^o)}, \qquad \delta(\mathsf{Dis}) = \frac{\mathsf{Dis}(CD_1^n,...,CD_L^n) - \mathsf{Dis}(CD_1^o,...,CD_L^o)}{\mathsf{Dis}(CD_1^o,...,CD_L^o)},$$
$$\delta(\mathsf{Div}) = \frac{\mathsf{Div}_{C_j}(CD_j^n) - \mathsf{Div}_{C_j}(CD_j^o)}{\mathsf{Div}_{C_j}(CD_j^o)}.$$

Observe that $\delta(\mathsf{Cov})$ and $\delta(\mathsf{Div})$ are defined in terms of the cluster CD being modified, whereas $\delta(\mathsf{Dis})$ is defined in terms of the entire clustering CDs. The improvements may be positive, negative or zero, and can have arbitrary magnitude. We are interested in non-negative and large ones. Experiments show that relative improvement is more advantageous than absolute improvement.

The *CDD measure* is defined in terms of the three factors. For the old CD $CD_1^o, ..., CD_L^o$ and new CD $CD_1^n, ..., CD_L^n$, the *CDD improvement* is defined by

$$\Delta\mathsf{CDD} = \begin{cases} \delta(\mathsf{Cov}) + \delta(\mathsf{Dis}) + \delta(\mathsf{Div}), & \text{if } min(\delta(\mathsf{Cov}), \delta(\mathsf{Dis}), \delta(\mathsf{Div})) \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Observe that in the formula we took the sum of the individual improvements for the three factors and insisted that each improvement is non-negative. We can also replace "sum" by "multiply", or drop the non-negative improvement requirement; however, experiments show that these do not perform as well.

When combining multiple factors to form a quality measure, trade-off among the factors occurs. In the above formula each factor carries a constant and equal weight; one may also use different and adaptive weights.

[5]Later we will consider adding or replacing one term only in our search.

## 4.6 The PagodaCD Search Strategy

We now consider how to efficiently construct succinct and informative CDs. We will present the PagodaCD [6] Algorithm, which is a layer-based replacement algorithm using the CDD surrogate quality measure.

### 4.6.1 Improvement-Based Replacement

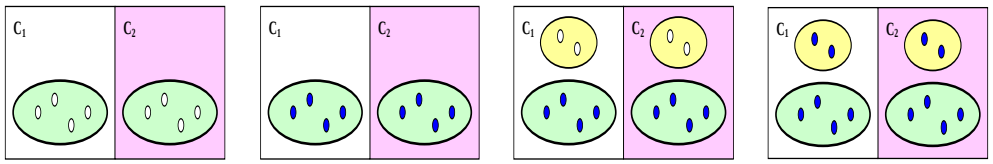
A natural but naive approach to searching good CDs is to repeatedly perform the best single-term replacement among all clusters and candidate terms, until no good replacement can be found. We call this *the basic improvement-based replacement* approach. Our experiments indicated that this method suffers from two drawbacks: it is still quite expensive, and it does not necessarily produce better CDs when the CD size increases. These drawbacks motivate us to introduce the PagodaCD Algorithm.

### 4.6.2 The PagodaCD Algorithm

Roughly speaking, our PagodaCD Algorithm divides the search process into multiple major steps, working in a layered manner. Each major step corresponds to the iterative selection of some $k_s$ new terms for each $CD_i$; it does not replace terms selected at earlier steps. This idea is illustrated using Figure 4.3. Here, the desired description size is 6, and $k_s = 4$ for the first major step and $k_s = 2$ for each subsequent major step. A shallow circle means that the term can still be replaced, whereas a filled circle means the term has been finalized and will not be replaced in the future. Initially, we select 4 terms per cluster and then iterate to find the best replacements among all terms for the current major step. In each subsequent major step, we add 2 more terms per cluster and then iterate to find

[6] A pagoda is a tower with multiple levels.

the best replacements of the newly added terms. The process continues until we finalize all 6 terms for each cluster. This process is level by level, and in each level all clusters are considered together. This is why the algorithm is called PagodaCD .




(a) (b) (c) (d)

Figure 4.3: Illustration of PagodaCD Algorithm When Description Size is 6

The PagodaCD Algorithm is described below.

---

**Algorithm 4.6.1** The PagodaCD Algorithm

Inputs*: Clusters $C_1, ..., C_L$; $k$ (CD size); baseSize, incSize, minImp;*

Outputs*: CDs*

Method*:*

1. *For each $i$, set $CD_i$ to $\emptyset$, and let $CP_i$ consist of the most frequent $50 + k$ terms occurring in cluster $C_i$;*
2. IterReplace*($\overline{CP}$,$\overline{CD}$,minImp,baseSize); //$\overline{CP}$ and $\overline{CD}$ are vectors*
3. *For $j = 1$ to $\frac{k-baseSize}{incSize}$* IterReplace*($\overline{CP}$,$\overline{CD}$,minImp,incSize);*
4. *Return $(CD_1, ..., CD_L)$* ■

---

Parameter *baseSize* is the number of terms to be obtained for each $CD_i$ in the first major step, and *incSize* is the number of terms to be added for each $CD_i$ in each subsequent major step. Parameter *minImp* is a user given minimum quality improvement threshold.

The IterReplace procedure is described below. It is used to select *stepSize* new terms for each $CD_i$, while keeping the terms selected in previous levels unchanged. It first selects the most frequent *stepSize* unused terms from the candidate term pools, and then use the $CDD$ measure to repeatedly select the best replacement terms. For each iteration, it finds the best replacement term among all clusters and all terms for the current major step. This is repeated until no replacement term with significant quality improvement is found.

---

**Algorithm 4.6.2** IterReplace($\overline{CP}$, $\overline{CD}$, *minImp*, *stepSize*)

*// $\overline{CP}$ is vector of candidate term pools $CP_1, ..., CP_L$;*

*// $\overline{CD}$ is vector of CDs $CD_1, ..., CD_L$;*

*// These two vectors are passed by reference;*

1. *For each $i$, let $CDTL_i = \{$the most frequent stepSize of terms in $CP_i - CD_i\}$, and let $CD_i = CD_i \cup CDTL_i$;*

2. *Repeat until no replacement is found:*

   *- For each $i$, term $t^o \in CDTL_i$ and term $t^n \in CP_i - CD_i$, compute $\Delta CDD_i(t^o, t^n)$ for the hypothetical replacement of $t^o$ in $CD_i$ by $t^n$. Suppose the best replacement among all possible ($t^o$,$t^n$) pairs for $i$ is $\Delta CDD_i(t_i^o, t_i^n)$, achieved at $t_i^o$ and $t_i^n$.*

   *- Let $C_j$ be the cluster with the largest $\Delta CDD_i$, i.e.*
   *$\Delta CDD_j(t_j^o, t_j^n) = \max_i \Delta CDD_i(t_i^o, t_i^n)$.*

   *If $\Delta CDD_j(t_j^o, t_j^n) > minImp$, then (a) let $CD_j^n$ (respectively $CDTL_j$) be the result of replacing $t_j^o$ in $CD_j$ (respectively $CDTL_j$) with $t_j^n$, (b) replace $CD_j$ by $CD_j^n$, and (c) keep the other $CD_i$ unchanged.* ■

---

Notice that PagodaCD uses IterReplace to do the replacement only in a local one-layer-at-a-time manner. This leads to both faster computation and the monotone-

quality behavior (getting higher F-scores when CDs become larger). We also note that one can use other measures in the place of the CDD measure.

We now give a complexity analysis on the PagodaCD Algorithm for constructing CDs. Let $L$ denote the number of clusters, $\gamma$ the average number of candidate terms for a cluster, $k$ the desired size of the CD for a cluster, $|D|$ the total number of documents, and $\tau$ the number of unique terms. The space complexity is $O(\tau|D|/8+\tau)$. (We represent a document as a bit vector, and we keep the bit vectors for all documents and a vector of all terms in main memory for fast access.) The time complexity is $O(\rho Lk\gamma|D|/32)$, where $\rho$ is the total number of single-term replacements actually performed which also depends on the $minImp$ threshold. Notice that $Lk\gamma|D|$ is the cost of finding the best single-term replacement for the given CDs.

### 4.6.3 Preselecting Candidate Terms

We conclude this section with some remarks on preselection of candidate terms. For large document collections, the number of unique terms can be very large. Constructing CDs from all those terms is expensive. Moreover, some terms will not contribute much to quality CDs, especially when some terms only been appeared in few documents. To address these concerns, it is desirable to select and use only a subset of terms for constructing the CDs. In this paper, we preselect a number of the most frequent terms for each $C_i$ as candidate terms. Notice that the choice of the number of candidate terms involves a trade-off between quality and efficiency. Here, we choose to have that number be $\gamma = 50 + k$, where $k$ is the desired description size (or CD size) for each cluster.

Another way is the *DFD-based preselection* approach, where we consider, in addition to the document frequency of terms in a given cluster, the discriminating

power of terms against other clusters. For each term $t$ in a cluster $C_i$, let

$$DFD_i(t) = \begin{cases} -1, & \text{if } |\mathsf{INT}_{C_i}(t)| > \max_{j \neq i} |\mathsf{INT}_{C_j}(t)| \\ \frac{|\mathsf{INT}_{C_i}(t)|}{|C_i|} * \log_2(\frac{|\mathsf{INT}_{C_i}(t)|}{\sum_{j \neq i} |\mathsf{INT}_{C_j}(t)|+1}), & \text{otherwise.} \end{cases}$$

Observe that $\frac{|\mathsf{INT}_{C_i}(t)|}{|C_i|}$ is $t$'s frequency in $C_i$, and $\frac{|\mathsf{INT}_{C_i}(t)|}{\sum_{j \neq i} |\mathsf{INT}_{C_j}(t)|+1}$ is the ratio of $t$'s frequency in $C_i$ over the sum of its frequency in other clusters. So $DFD_i(t)$ is large if (1) $t$ is frequent in $C_i$ and (2) $t$ is infrequent in other clusters; condition (2) indicates that $t$ has high discriminatory power. The log function allows to balance frequency against discriminating power. The *DFD-based selection procedure* picks the $\gamma$ top terms for each $C_i$, ordered by decreasing $DFD$ value.

## 4.7 The CumulativeCD Search Strategy

We now turn to the *CumulativeCD* algorithm. This is an additive method; it uses a greedy forward search strategy by considering one cluster at a time (the outer loop), and for each cluster, by adding one term (the best term for the cluster) at a time (the inner loop). Once a term is selected it will never be replaced.

This idea can be illustrated using Figure 4.4. Here, the desired description size is 6, and a filled circle means the term has been selected and can not be replaced in the future. At the beginning, we select one term for the first cluster. We then add one-term-a-time for this cluster until we get all 6 terms. This process will be continued for the next cluster until all clusters are considered.

The *CumulativeCD* algorithm is described in Algorithm 4.7.1. Let $M$ be a quality measure.

---

**Algorithm 4.7.1** The *CumulativeCD* algorithm

Inputs*: Clustering $C_1, ..., C_L$; description size $k$;*

Outputs*: CDs for the clustering*

Method*:*

1. *Select candidate pool $CP_i$ for each cluster; initialize $CD_1, ..., CD_L$ to {};*
2. *Repeat for each cluster $C_i$:*
   - *Repeat until $CD_i$ has $k$ terms:*
     - *Select the first term and add it to $CD_i$;*
     - *For each term $t^n \in CP_i - CD_i$, compute $\Delta M(t^n)$ for the hypothetical addition of $t^n$ to $CD_i$;*
     - *Suppose the best addition $\Delta M(t')$ is achieved at $t'$. Add $t'$ to $CD_i$;*
3. *Return all CDs* ■

---

When computing $\Delta M(t^n)$, we use all computed $CD_1, ..., CD_L$ and we use $CD_i \cup \{t^n\}$ as the hypothetical new $CD_i$. Again the choice for the preselection method in step (1) can be either frequency or DFD based.

The *CumulativeCD* algorithm is very efficient compared with *PagodaCD*, and its performance is very competitive when used with the DFD candidate preselection method.

We now give a complexity analysis on the *CumulativeCD* algorithm for constructing CDs. Let $L$ denote the number of clusters, $\gamma$ the number of candidate terms for a cluster, $k$ the desired size of the CD for a cluster, $|D|$ the total number of documents, and $\tau$ the number of unique terms. The space complexity is $O(\tau|D|/8+\tau)$, similar to the *PagodaCD* algorithm. The time complexity is roughly

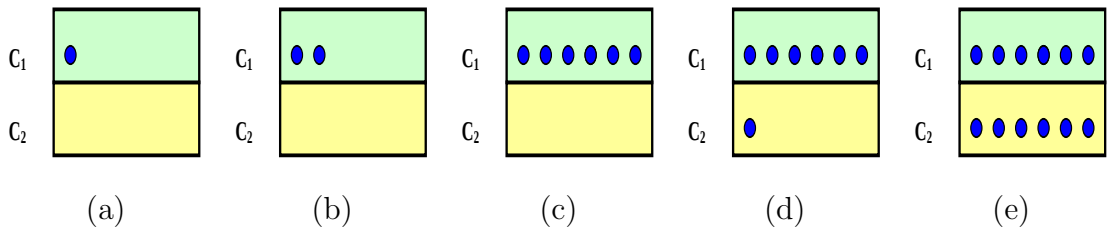



Figure 4.4: Illustration of *CumulativeCD* Algorithm When Description Size is 6

$O(Lk\gamma|D|)$. Notice that we need to use $\gamma|D|$ operations to find the best next term, there are a total of $Lk$ terms to select, and there is no replacement.

## 4.8 Experimental Evaluation

In this section, we present an empirical evaluation of various CD construction algorithms, including ours. The goals of the experiments are (1) to demonstrate the superior quality of CDs produced by our algorithms than those produced by other algorithms, and (2) to validate the claims that *coverage*, *disjointness* and *diversity* are important factors for constructing succinct and informative CDs.

### 4.8.1 Experiment Setup

In this thesis, we only consider CDs and assume that a clustering is given by other algorithms. We used the *vcluster* command of the *Cluto* [63] toolkit to generate the clusterings; the clustering algorithm we used is *repeated bisecting*, which was shown to outperform the basic k-means and UPGMA algorithms [115]. Below, all data sets are divided into 10 clusters, unless indicated otherwise.

We evaluate the following CD construction approaches, in addition to PagodaCD. The "Descriptive CD" and "Discriminating CD" approaches were described in Section 4.2, and were generated using the *Cluto* package. The "Frequency-based CD" were simply the most frequent terms from each cluster. Finally, the "COBWEB-like CD" approach is also considered, which uses the utility category [10, 33, 38] as the search criterion and uses our PagodaCD strategy to search.

All experiments were conducted on a PC with 2.4GHZ CPU and 512MB RAM, running Windows XP. All programs were coded in *Java*.

| Data sets | # of docs | # of terms |
|---|---|---|
| Reuter2k | 2000 | 9660 |
| Reuter4k | 4000 | 14044 |
| Reuter8k | 8000 | 20274 |
| Reuter10k | 10000 | 22756 |

Table 4.2: Summary of Test Data Sets

### 4.8.2 Data Sets

Our experiments were performed on the Reuters-21578 [75] documents collection, which has been widely used by researches in the field of document classification and clustering.

The collection contains 21578 news articles, distributed in 22 files. We constructed five subsets, *Reuter2k*, *Reuter4k*, *Reuter6k*, *Reuter8k* and *Reuter10k*, containing 2k, 4k, 6k, 8k and 10k documents respectively, in the following manner: The 22 files were first concatenated in the order given. We then eliminated those documents with the following tags: TOPIC="BYPASS", LEWISSPLIT="NOT-USED" and TEXT TYPE="BRIEF" or "UNPROC"; such documents were often ignored by other researchers, since most of them contain little or no meaningful textual content. Finally, we got the desired number of documents from the concatenation starting from the beginning, i.e. *Reuter2k* contains the first 2000 documents from the concatenation, *Reuter4k* the first 4000 documents, and so on. All documents were preprocessed by removing stop-words and stemming words to their root forms, following common procedures in document processing. The final data sets have between 9660 and 22756 unique terms (see Table 4.2).

### 4.8.3 CD Quality

PagodaCD **vs.** ***CumulativeCD.*** We compared those two approaches over few dataset, and found that CDs produced by the PagodaCD algorithm are usually better than those produced by the *CumulativeCD* algorithm. However, *CumulativeCD* approach is much fast than the PagodaCD approach. In the following experiments, we choose PagodaCD as our representative approach.

PagodaCD **vs. Other Existing Approaches.** We compare the CD quality of thePagodaCD with other existing approaches. Figure 4.5 shows the average F-score of different approaches for different CD-Sizes in the Reuter8k data set. We can see that PagodaCD outperforms the *Descriptive* approach, which is the best among others, by at least 15% relative (or 8% absolute) percent for all description sizes. Figure 4.6 shows the average F-score of different approaches in Reuter2k, 4k, 6k, 8k and 10k data sets, with the description size fixed at 8. Again, the PagodaCD Algorithm outperforms the *Descriptive* approach by at least 10% relative (or 7% absolute) percent. For other data sets and description sizes, the performance comparison is similar.

Interestingly, when the description size increases, the average F-score of CDs produced by PagodaCD and *COBWEB-like CD* also increases. However, this is not true for other approaches. Figure 4.5 indicates that the F-score of other approaches jumps up and down, and it even deteriorates in some cases when the description size increases.

Table 4.3 shows some description terms produced by different approaches in Reuter4k when the description size is 4. We selected 2 clusters from total of 10 clusters to save space. Although terms are in their root or abbreviated form, we can still sense that cluster 4 is about "large-scale" bank financing and cluster 7 is about stocks. This will be more obvious to domain experts. PagodaCD and *Descriptive* CDs give us better sense about these topics. For *Discriminating* CDs, there are

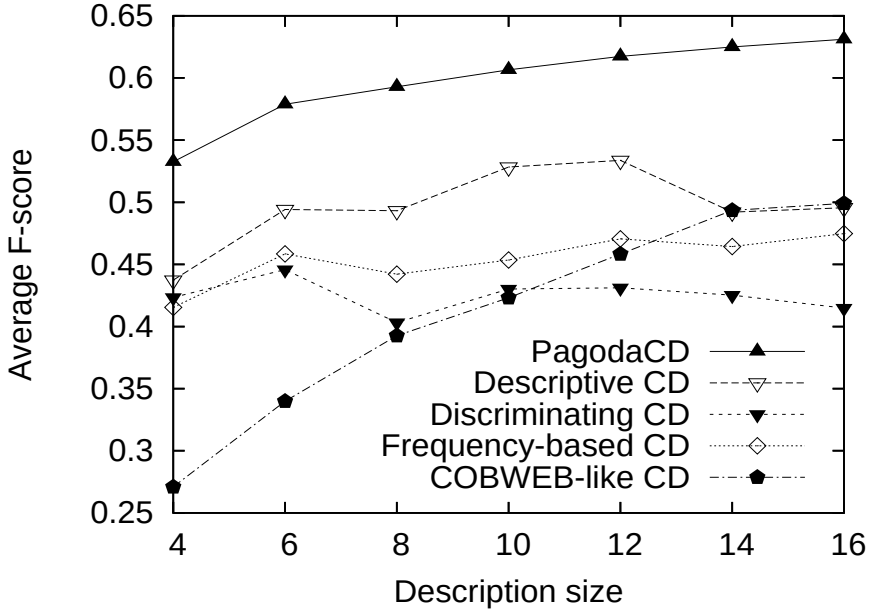



Figure 4.5: F-score vs CD-Size in Reuter8k.

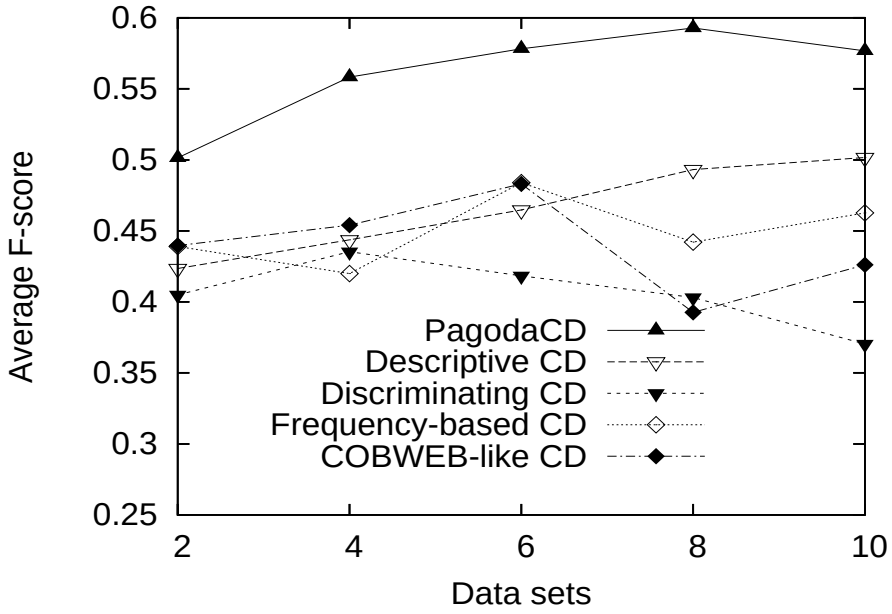



Figure 4.6: F-score vs Data Sets (2k, 4k, 6k, 8k, 10k) When CD-Size = 8.

| **Approaches** | **Cluster 4** | **Cluster 7** |
|---|---|---|
| PagodaCD | bank pct financ billion | offer dlr stock share |
| Descriptive CD | bank rate stg debt | share offer stock common |
| Discriminating CD | bank net shr loss | share net shr offer |
| Frequency-based CD | bank said pct rate | share dlr inc compani |
| COBWEB-like CD | funaro reschedul imf citibank | registr redeem subordin debentur |

Table 4.3: CDs by Different Approaches When CD-Size = 4 in Reuter4k.

duplicated terms in both clusters, namely *net* and *shr*. For *Frequency-based* CD, *inc* and *compani* in cluster 7 give redundant information.

**Impact of Clustering Quality on CD Quality.**

Clustering quality has big impact on CD quality. High clustering quality means that documents in a cluster are very similar to each other, but are very different from those in other clusters. It turns out that CDs constructed from high quality clusterings tend to have high quality, and those constructed from low quality clustering tend to have low quality. To demonstrate the effect of clustering quality, we produced different clusterings (5, 10, 15, 20-way) from Reuter4k. We measured the clustering quality by the weighted sum of the difference between the internal similarity and external similarity of each cluster. Interestingly, the clustering quality deteriorates when the number of clusters increases for this dataset. Figure 4.7 indicates that CD quality also deteriorates when clustering quality deteriorates.

We measured the clustering quality by the weighted sum of the difference between the internal similarity and external similarity of each cluster, namely

$$\frac{1}{L}\Sigma_{i=1}^{L}\frac{|C_i|}{|D|}(ISim_i - ESim_i),$$

where $C_1, ..., C_L$ are the clusters and $D = \cup_{i=1}^{L} C_i$, $ISim_i$ and $ESim_i$ are respectively the internal and external similarities of $C_i$. The *Cluto* toolkit can produce

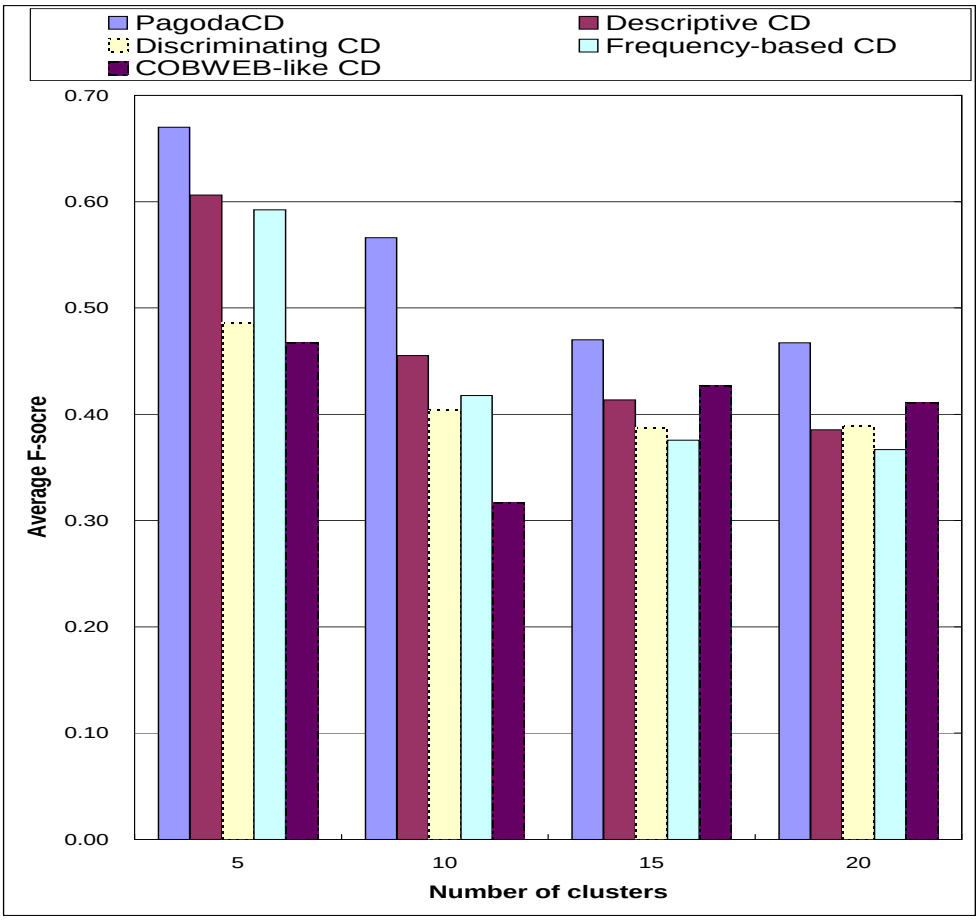



Figure 4.7: Average F-score vs. Number of Clusters in Reuter4k.

| CID | Size | ISim | ISdev | ESim | ESdev |
|---|---|---|---|---|---|
| 0 | 170 | 0.295 | 0.085 | 0.015 | 0.006 |
| 1 | 520 | 0.274 | 0.089 | 0.015 | 0.006 |
| 2 | 896 | 0.052 | 0.017 | 0.017 | 0.005 |
| 3 | 825 | 0.031 | 0.011 | 0.012 | 0.005 |
| 4 | 1589 | 0.022 | 0.009 | 0.014 | 0.007 |

Table 4.4: Internal and External Similarities for 5-ways Clustering in Reuter4k.

| # Clus | WA. ISim | WA. ESim | ISim-ESim |
|---|---|---|---|
| 5 | 0.0150 | 0.0029 | 0.0121 |
| 10 | 0.0098 | 0.0017 | 0.0081 |
| 15 | 0.0073 | 0.0011 | 0.0061 |
| 20 | 0.0059 | 0.0009 | 0.0051 |

Table 4.5: Weighted Average (WA.) of Similarities vs Number of Clusters in Reuter4k.

those $ISim_i$ and $ESim_i$.

As an example, Table 4.4 shows the similarities for a 5-way clustering. *CID* is the cluster id, *Size* the number of documents in a cluster, *ISim* the average similarity among the objects of a cluster (i.e., the internal similarity), *ESim* the average similarity between the objects of a given cluster and the objects in other clusters (i.e., external similarity), and *ISdev* and *ESdev* are the standard deviation of the average internal and external similarities, respectively.

Table 4.5 shows the weighted average of similarities for different clusterings. "ISim-ESim" measures the difference between the weighted average of internal and external similarities for all clusters in each given clustering. Notice that clustering

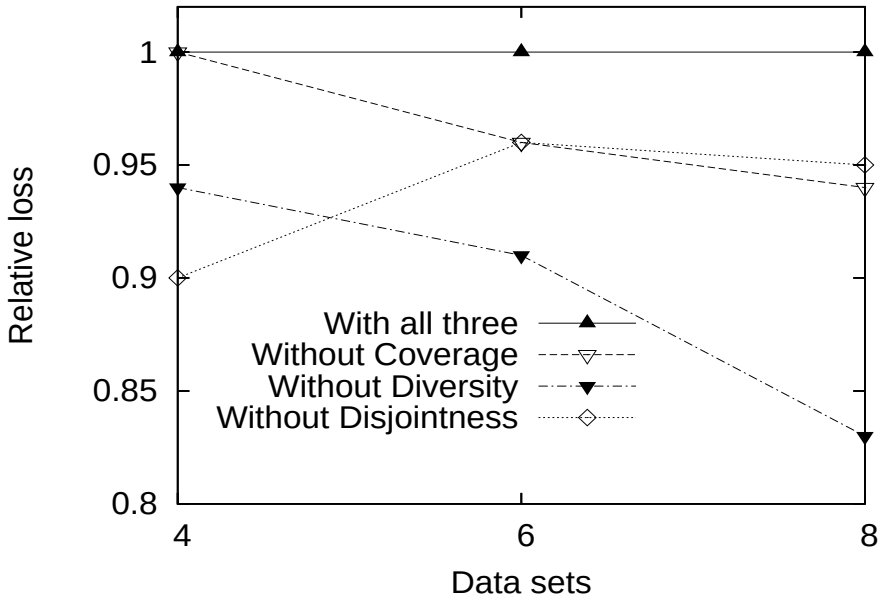



Figure 4.8: Relative F-score Loss vs Data Sets (4, 6, 8k) When One Factor is Left Out.

quality deteriorates when the number of clusters increases.

### 4.8.4 Importance of the Three Factors

Experiments confirmed that the three factors of *coverage*, *disjointness* and *diversity* are very important for constructing informative CDs. Indeed, if we leave any of them out, the CD quality is not as good as when all three are used. Figures 4.8 and 4.9 show the importance of different factors in terms of relative loss or gain of average F-score. Because the candidate terms are frequent terms, *coverage* is less important than *diversity*. In other experiments we observed that, when *coverage* is less important, the other two factors, especially *diversity*, are very important.

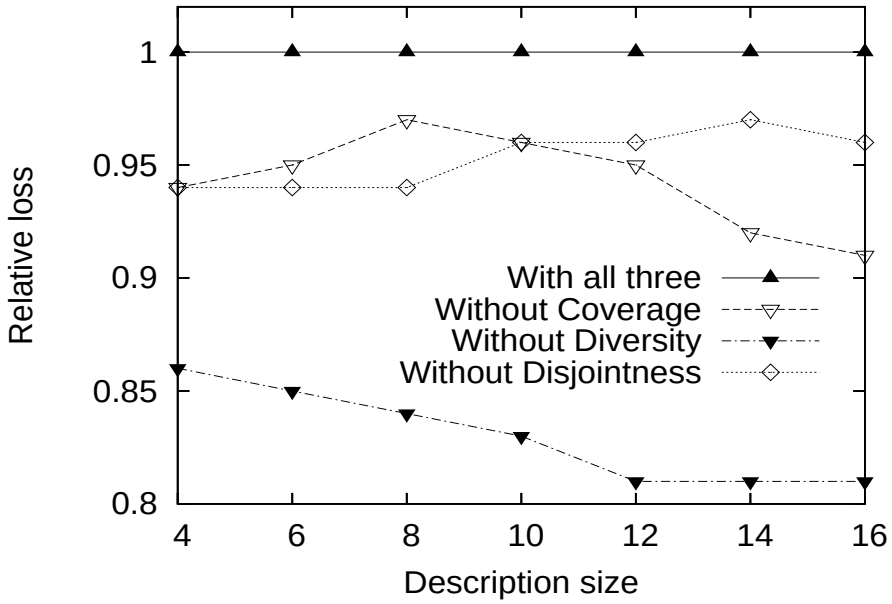



Figure 4.9: Relative Loss vs CD-Sizes When One Factor is Left Out in Reuter8k.

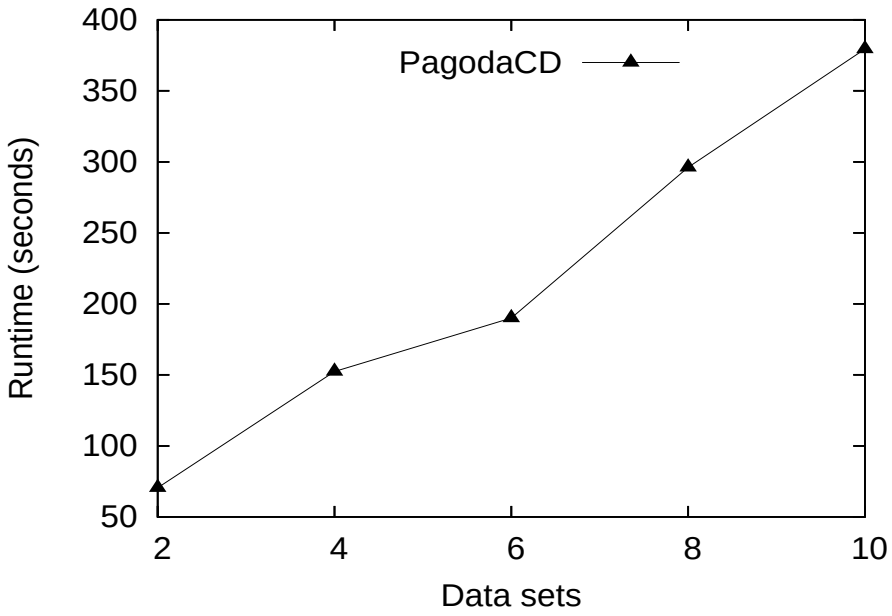



Figure 4.10: Runtime vs Data Sets (2k, 4k, 6k, 8k, 10k) When CD-Size = 8.

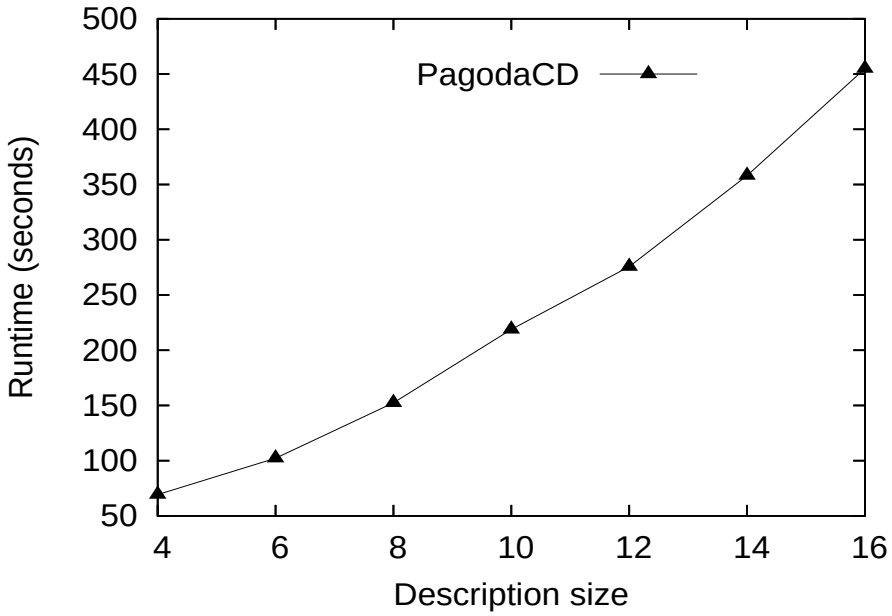



Figure 4.11: Runtime vs CD-Size in the Reuter4k Data Set.

### 4.8.5 Computation Time

Figure 4.10 shows the computation time used by the PagodaCD Algorithm, for different data sets when the description size is 8, and Figure 4.11 shows that for different description sizes for the Reuter4k data set. We can see that the execution time increases in a linear manner as the description size or number of documents increases.

### 4.8.6 Other Ways to Combine the Three Factors

We also considered combining the improvements of the three factors using products, in stead of using sum. However, the results are not as good as using sum.

## 4.9 Constructing CDs Using Genetic Algorithm

In this dissertation, we also try to construct CDs using the genetic algorithm (GA). We will give our GA settings and some preliminary experimental results in the following subsections.

### 4.9.1 Introduction

A genetic algorithm (or GA) is a search technique used in computing to find true or approximate solutions to optimization and search problems [129]. Genetic algorithms, categorized as global search heuristics, are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination).

A typical genetic algorithm requires two things to be defined: one is a genetic representation of the solution domain, and another one is a fitness function to evaluate the solution domain. A standard representation of the solution is as an array of bits. A pseudo-code GA algorithm is described in Algorithm 4.9.1.

---

**Algorithm 4.9.1** The Typical Genetic Algorithm

1. *Choose initial population*
2. *Evaluate the fitness of each individual in the population*
3. *Repeat until <terminating condition> is met*
    - *Select best-ranking individuals to reproduce*
    - *Breed new generation through crossover and mutation (genetic operations) and give birth to offspring*
    - *Evaluate the individual fitness of the offspring*
    - *Replace worst ranked part of population with offspring* ■

GA has been used successfully by many researchers in variety of applications [29, 36, 88], and it has also been successfully used for feature selection [65, 133, 91]. Although CDs and features are different in many perspectives, we believe that the GA techniques can also be used to construct CDs.

### 4.9.2 Genetic Algorithm (GA) Settings

In this work, we use the basic GA with the following settings. Note that the CDs are constructed in a cluster-by-cluster fashion, one cluster at a time. Let $N$ be the base population size and $k$ be the desired number of items in each individual (i.e. description size for each cluster).

- **Representation**: integer. Basically, each chromosome is made up of $k$ integers, and each integer represents a particular term that occurs in the document collection. Note that in the real implementation, a chromosome is made from $k$ objects, and each object contains the term's index (an integer) and their document frequency information.

- **Fitness**: F-score. We consider the $k$ terms in each individual as a query set with logical OR relationship among the terms. The "retrieval" quality is then measured by using the F-score which was described in Section 4.4.2.

- **Initialization**: creates a population of randomly initialized $N - 1$ individuals, and each individual has $k$ terms. We also make sure that the most frequent $k$ terms for the current-processing cluster are included as one individual. So there are totally $N$ individuals.

- **Parent selection**: random. Randomly select two different parents for mating.

- **Recombination**: one-point crossover. We also have the option for two-point crossover and uniform crossover.
- **Mutation**: probability of $1/k$ to mutate for each term in an individual. We also have the option for the probability of $2/k$.
- **Survivor selection**: generational. In each generation, $3 * N$ offsprings are generated. We keep the fittest individual in the previous generation. So, $(3 * N + 1)$ individual will compete for survival based on their fitness, and the fittest $N$ individuals among them will survive.

### 4.9.3 Preliminary Results

#### 4.9.3.1 Datasets

To verify the effectiveness of the GA on constructing CDs, we conduct preliminary experiments on subsets of the Reuters-21578 documents collection. Most of the experiments were performed on the dataset *Reuter1k*, in which 1000 documents were selected from the beginning of the collection. We ignore those documents with little or no meaningful textual content, and we also perform stop-words removal and stemming on those documents. The final dataset has 6673 unique words. We also use the *Reuter2k* dataset described in Section 4.8.2.

The datasets are further clustered into 10 clusters using the *Cluto* [63] toolkit. The number of documents in each cluster are listed in Table 4.6. We can see that the last few clusters have more documents than the others. This is because the clusters are numbered in terms of the cluster quality, so the later clusters contain more diversified topics.

Usually, the number of terms we want to use to describe the cluster is small; a practical choice will be less than 20. However, the number of unique terms to choose from is often very large, for example 6773 in *Reuter1k*. It will be very

| Clusters | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **# of doc in Reuter1k** | 57 | 81 | 57 | 55 | 66 | 73 | 119 | 152 | 114 | 226 |
| **# of doc in Reuter2k** | 119 | 161 | 128 | 92 | 112 | 185 | 154 | 355 | 189 | 505 |

Table 4.6: Number of Documents in Each Cluster

expensive to construct CDs from such large search space by using F-score as fitness function; furthermore, not all of the terms are useful to describe the clusters. In this project, we select 100 most frequent terms as candidates terms to search from.

#### 4.9.3.2 Performance of Benchmark Settings

The basic setting of the GA described in Section 4.9.2 is considered as benchmark setting for this experiment. In addition to that setting, the following parameters are also specified:

- Population size: 40
- Maximum generation: 80
- Random generator seeds: 14391

Here we only compare the CD quality between the GA approach and the frequent-term based approach. We do not compare with other approaches, as mentioned earlier in the chapter, which could be our future work.

From Table 4.7, we can see that the F-score of GA CDs is much better than the frequency-based CDs. For example, when $k = 8$, there is relatively 62.42% improvement for the GA method comparing to the frequency-based approach.

Note that when the number of terms increases in the CDs, the F-score of the frequency-based CDs is actually decreasing, while the GA approach is not. This is simply because more frequent terms will inevitably introduce more irrelevant

| CD Size | GA CD | Frequency-based CD | Relative Improvement(%) |
|---|---|---|---|
| **k=4** | 0.333651 | 0.154375 | 53.73 |
| **k=8** | 0.344427 | 0.129441 | 62.42 |
| **k=12** | 0.349188 | 0.124584 | 64.32 |
| **k=16** | 0.341655 | 0.119172 | 65.12 |

Table 4.7: The Weighted Average F-score for All 10 Clusters of Reuter2k

coverage for the CDs. On the other hand, the GA based approach try to find some other terms which can more precisely describe the cluster. Also notice that the overall F-score for both approaches are not that high, this is because the difficulty nature of the problem, and there is more room for improving.

Figure 4.12 shows the fitness (F-score) vs. the generation plot when $k = 2$ for the cluster 2. From the plot we can see that the fitness improved very dramatically during the first 20 generations, and has little improvement is the rest of the generations. In many different approaches (which will be discussed later) that we tried, this quick convergence trend remains to be mostly the same. This may have happened due to the nature of the problem, but the true cause is still unclear. For other $k$ and clusters, the performance trends are similar.

#### 4.9.3.3 Performance Based on Some Variations

The following variation has been made to test the impact of different factors. Note that each time we only change one factor and we let all other factors take the same values as the ones mentioned in the benchmark setting. Also notice that the difference in F-score is often small.

**Change The Maximum Number of Generations**

Figure 4.13 shows the performance improvement when changing the maximum

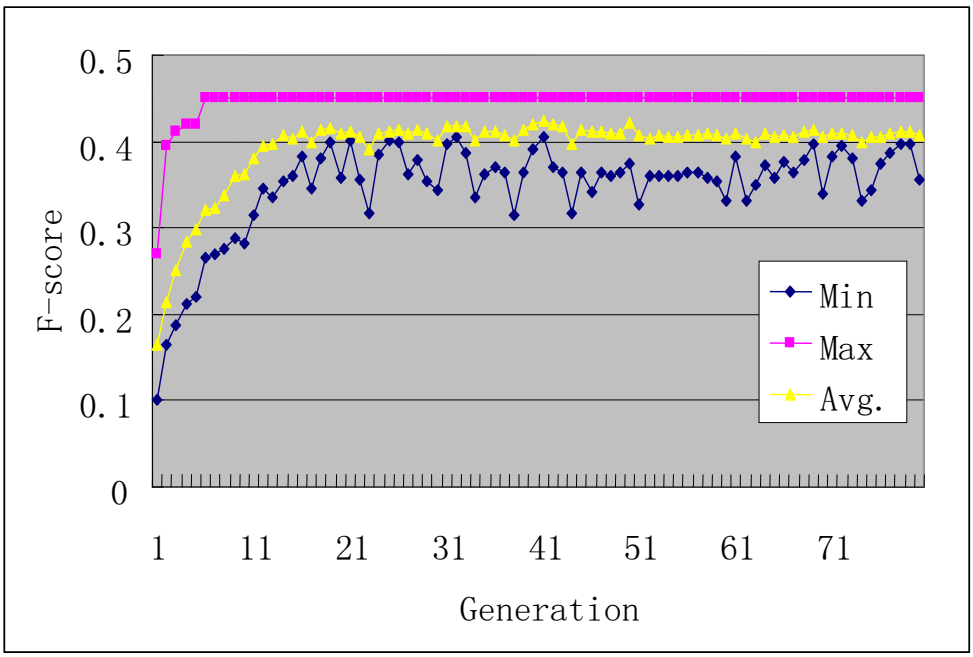



Figure 4.12: Fitness vs. Generation When k = 2 for the Cluster 2

number of generations (MG). Note that the horizontal axis is the number of terms in descriptors. The number of terms (CD size) is 4 times the number on the horizontal axis. i.e. 1 means 4 terms, 2 means 8 terms, and so on.

Clearly we can see that the performance is improving as the number of generation increasing, even though sometimes the changes are small.

**Change Population Size**

Table 4.8 shows the performance when the population size (PS) is changed. From the table we can see that the improvement of the performance is very limited. This is simply because we already allow lager number of generations in the setting.

Comparing these results with the results when the number of generations is changed, it looks like that the generation is more important (or more sensitive to the results) than the population size. Although the population size is also important, we definitely need to give the GA enough generations to evolve in

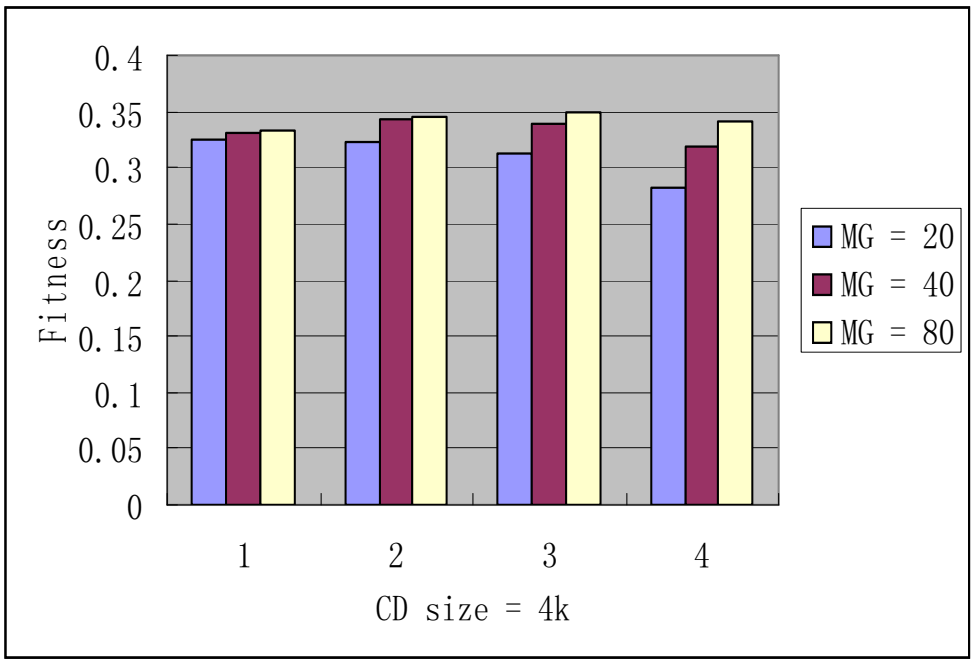



Figure 4.13: F-score for Different Maximum Number of Generations

| | **PS = 20** | **PS = 40** | **PS = 60** |
|---|---|---|---|
| **k=4** | 0.333697 | 0.333651 | 0.334173 |
| **k=8** | 0.344134 | 0.344427 | 0.347676 |
| **k=12** | 0.341146 | 0.349188 | 0.346892 |
| **k=16** | 0.333658 | 0.341655 | 0.340102 |

Table 4.8: F-score for Different Population Size

| | Generational | Elitism | Tournament |
|---|---|---|---|
| **k=4** | 0.333651 | 0.33487 | 0.328592 |
| **k=8** | 0.344427 | 0.348766 | 0.34397 |
| **k=12** | 0.349188 | 0.349144 | 0.344259 |
| **k=16** | 0.341655 | 0.341514 | 0.335392 |

Table 4.9: F-score for Different Survivor Selection Method

order to get the optimal solutions.

**Change Survivor Selection Method**

In Table 4.9, we give the performance when different survivor selection methods are used. Three survivor selection methods are *generational*, *elitism* and *tournament*. For the tournament selection, the tournament size is increased as the number of generation evolves. The tournament size range is from 2 (first generation) to $N/2$ (last generation). From these results, we can see that there is not much difference for these three methods in this particular setting.

**Change Candidate Size**

Figure 4.14 shows the performance when the number of candidate terms changes. From this figure we can clearly see some performance improvement when we have more terms choose from. However, more candidates will require more time for the GA to evolve. Also notice that when the CD size is large, for example $k = 16$, the performance improvement is more significant when the candidates size is large. This is because large candidates increase the diversity of the population.

**Change Recombination Methods**

Table 4.10 shows the performance when different recombination methods used. From this table we can see that there is no big different between one-point and two-point crossover, but it seems that both of them are better than the uniform crossover.

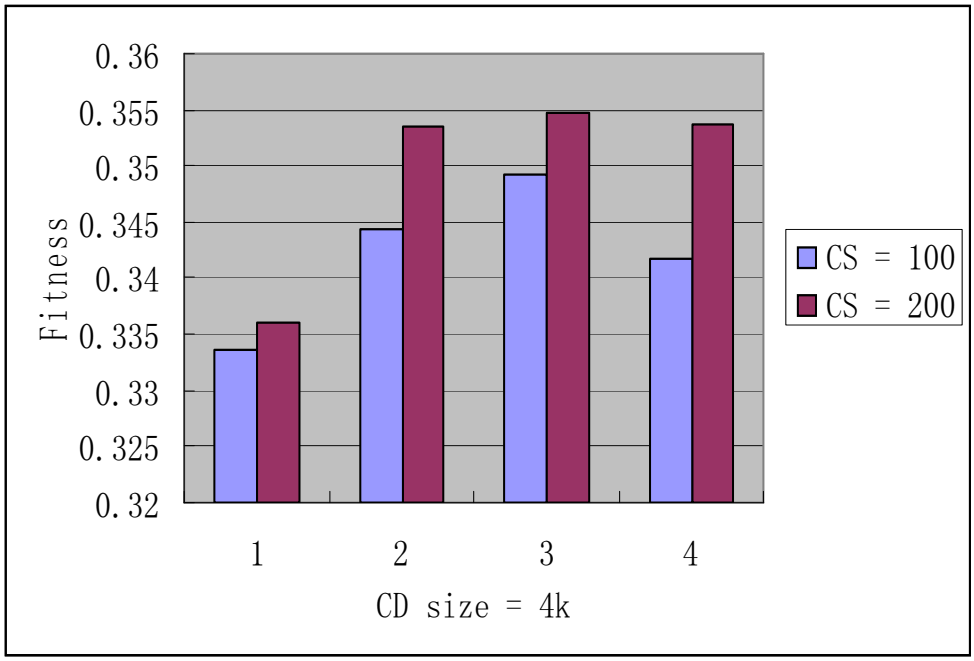



Figure 4.14: F-score When The Candidate Size(CS) Is 100 And 200

| | **One-point Crossover** | **Two-point Crossover** | **Uniform Crossover** |
|---|---|---|---|
| **k=4** | 0.333651 | 0.334093 | 0.332742 |
| **k=8** | 0.344427 | 0.347291 | 0.332742 |
| **k=12** | 0.349188 | 0.349293 | 0.348978 |
| **k=16** | 0.341655 | 0.340784 | 0.337127 |

Table 4.10: F-score for Different Recombination Methods

| | MP = 1/k | MP = 2/k |
|---|---|---|
| **k=4** | 0.333651 | 0.326639 |
| **k=8** | 0.344427 | 0.327632 |
| **k=12** | 0.349188 | 0.323707 |
| **k=16** | 0.341655 | 0.31031 |

Table 4.11: F-score for Different Mutation Probabilities (MP)

| CD Size | GA CD | Frequency-based CD | Relative Improvement(%) |
|---|---|---|---|
| **k=4** | 0.347259 | 0.167179 | 51.86 |
| **k=8** | 0.35222 | 0.141383 | 59.86 |
| **k=12** | 0.349511 | 0.131247 | 62.45 |
| **k=16** | 0.338679 | 0.127422 | 62.38 |

Table 4.12: The Weighted Average F-score for All 10 Clusters of Reuter2k

**Change Mutation Probability**

Table 4.11 shows the performance when the probability of mutation (MP) is different. We can see that the smaller mutation probability seems better than the larger probability in this setting. The reason for this may be because too many mutations will destroy some good subunits which have been learned in the past. This is similar to the reason for the poor performance of the uniform crossover mentioned above.

**Change Data Sets**

Table 4.12 shows the performance when the *Reuter2k* dataset is used for the benchmark setting. Clearly again, we can see that GA approach is much better the frequency-based approach. The effects of changing other factors are similar to those discussed above for the*Reuter1k* dataset.

**Other Factors**

During the experiments, we also tried other factors, such as whether or not to allow duplicated terms in CDs, random generator seeds, etc. Some preliminary experiment results show that no big difference on those variations.

### 4.9.4 Conclusions and Future Works

In this dissertation, we attempted to use the well-studied genetic algorithm (GA) for constructing CDs. We used the standard GA in the experiment. The CD quality was measured using F-score, which was also used as fitness function. We also examined the effects of different factors on the performance of GA, such as population size, maximum number of generations, recombination methods, survivor selection methods, mutation probability, etc.

The experiment results on subsets of Reuters collection demonstrated that CDs produced by GA are much better than the frequency-based approach in terms of F-score. Therefore, we believe that GA is a very promising tool to construct succinct CDs for large document repositories, and it deserves our further attention.

Although we considered different factors separately in the experiment, we have not considered changing multiple factors simultaneously, which will be one of our future works. In addition, we would also like to use our CDD measure as fitness function, and also compare the GA approach with other available approaches, including PagodaCD, in our future work.

## 4.10 Summary

In this chapter, we have studied the problem of CDs extensively for any given document clustering, regardless of the clustering algorithm used to produce the clusters. We argued that constructing succinct and informative CDs is an im-

portant component of clustering process, especially for managing large document repositories. We believe that succinct and informative CDs can help users quickly get a high-level sense of what the clusters contain, and hence help users use and "digest" the clusters more effectively.

We discussed and formalized how to interpret the CDs and how to resolve perception competition. We introduced a *CD-based classification* approach to systematically evaluate CD quality. We identified a surrogate quality measure, the CDD measure, for efficiently constructing informative CDs. We gave a layer-based replacement search method called PagodaCD and a greedy forward search method called *CumulativeCD* for constructing CDs. Experimental results demonstrated that our method can produce high quality CDs efficiently, and CDs produced by PagodaCD also exhibits a monotone quality behavior.

In this chapter, we also presented our efforts on using genetic algorithm (GA) to construct CDs. The preliminary experimental results suggested that GA is a very promising tool to construct succinct CDs for large document repositories. We plan to devote more time on this approach in our further work.

Our work in this chapter can be applied to any document repositories and clustering algorithm. However, we believe that the quality of this work can be improved if we can do the following: (1) performing clustering and constructing informative CDs at the same time in order to get high quality CDs and clusterings, (2) giving the three factors different weights in different situation, and considering new surrogate quality measures, (3) considering synonyms and taxonomy in forming CDs, (4) involving human evaluation efforts to further validate the understandability of CDs, and (5) adapting previous ideas on the use of emerging patterns and contrasting patterns for building classifiers [26, 27, 76, 45] to construct succinct and informative CDs.

In the next chapter, we will present our approach to perform clustering and

constructing CDs at the same time for e-rulemaking feedback repositories (ER-FRs). To improve the understandability and descriptiveness of CDs, we will also add more information to the CDs, such as some statistical information.

# Chapter 5

# Clustering of ERFR with Simultaneous Construction of Succinct Cluster Descriptions

Succinct and informative cluster descriptions (CDs) are useful for managing large e-rulemaking feedback repositories (ERFR). In Chapter 4, we studied how to interpret the CDs, how to evaluate CD quality and how to construct high quality CDs for given document clusterings. However, as mentioned in Chapter 4, previous clustering algorithms mainly focused on cluster formation, and paid little attention to producing good CDs.

In this chapter, we will introduce our approach which deals with both clustering quality and CD quality at the same time – our approach will perform clustering and construct succinct CD simultaneously for any given ERFR. In addition, we also take consideration of those three important aspects (namely issues, stakeholders and opinions), which have been identified in Chapter 3, for given ERFR. We will see later that our approach can produce high quality summaritive digest (SD), which can serve as an informative navigation aid to rule-writers and analysts for

managing and digesting the underlying feedbacks more effectively.

## 5.1 Introduction

To help the rule-writers and analysts manage and digest large ERFR, researchers have been actively working on solutions to address those new challenges posted by ERFR in recent years. For example, there have been studies on duplicate or near-duplicate detection [132], cluster labeling [67, 17], text analysis in various ways [120], etc. Some other works were also discussed in Section 2.1.

When facing large amount of feedbacks (e.g. millions) for a proposed rule, we believe that, as also mentioned in earlier chapters, rule-writers and analysts probably want to ask the following fundamental questions : 1) what are the important issues that the public is concerned with? 2) which groups (i.e. stakeholders) are more concerned about these issues? 3) what are their opinions? 4) what are the typical arguments to support their opinions? etc. In addition, it is highly desirable to have good organization structure and informative description of the feedbacks, for the rule-writers and analysts to digest the feedbacks more easily. Recall that one of the goals of this dissertation is to address such challenges and needs.

In this chapter, we will introduce our approach to organize and summarize those feedbacks based on the knowledge of three important aspects of ERFR. The three aspects are opinions (O), issues (I) and stakeholders (S), which have been studied in Chapter 3. We propose to construct clusterings based on different combinations of those aspects. A clustering can be either flat or hierarchical. Different clusterings can meet different users' needs. A "best" clustering could be recommended to the user based on given goodness measure. We call our approach the *OIS-based* approach, even though the order of O, I and S may not be the order for the best clustering. The *OIS-based* approach can also generate a flat clustering,

which is like the traditional content-based approach but it uses the O, I and S as factors. In the meantime, we also construct succinct cluster descriptions (SCDs) for each cluster. The SCD consists of a set of key terms or phrases, which is similar to the CD discussed in Chapter 4; it also has some statistics information, which can be considered as enhancement to the CDs.

Roughly speaking, we first identify O, I and S for the given ERFR using the approaches discussed in Chapter 3. Then, we perform clustering and construct SCD simultaneously by incorporating those identified O, I and S. The O, I and S will be used in both feature selection and SCD construction. In addition, we will also select representative arguments (RAs) for each cluster in the clustering, which will be discussed in Chapter 6. Collectively, the clustering scheme, SCD and RAs form the summaritive digest (SD) for ERFR.

With the SD, rule-writers or analysts can easily grasp the main picture of how the public feel about the proposed rules. This kind of clustering structure and the SCDs provide users[1] a "virtual map" to navigate those feedbacks that they are interested in, and can help users to digest the ERFR. Again, users can view a different clustering by using different order of O, I and S when the clustering is generated, depending on their preference and on how they want to see the summary.

Our approach is different from traditional text clustering and summarization in a number of ways. First of all, our hierarchical clustering is based on some predefined dimensions (i.e. O, I and S) rather than "general-content based". Second, we are mainly interested in features related to O, I and S, because we try to answer the question: "who cares about what, and how?" in the context of managing ERFR. Our feature selection also treats the proposed rule as background knowledge. Third, we perform clustering and construct SCD at the same time. Finally,

[1]Here, users are rule-writers or analysts. We use them interchangeably in this thesis.

we construct succinct and informative summarization (containing SCDs and RAs) for each cluster based on those predefined dimensions. We do not summarize the feedbacks by rewriting the original sentences, but we do select some representative sentences for each cluster.

In the following sections, we will first survey some related works. Then we will introduce our clustering approach, followed by our SCD construction approach. Even though those two steps will be performed at the same time, we separate them into different sections for clarity. We will also discuss our approaches to evaluate clustering quality and SCD quality, and show some experimental results. At the end, we will give a short summary for this chapter.

## 5.2 Related Works

Related works can be divided into the following three groups: document clustering, document summarization and the application of e-rulemaking. We will discuss each of them below.

**Document Clustering.** As mentioned in Section 2.2, there have been many studies on document clustering during the past several decades. In general, clustering approaches can be categorized as *agglomerative* or *partitioning* based on the underlying methodology of the algorithm, or as *hierarchical* or *flat* (non-hierarchical) based on the structure of the final solution. One example of the partitioning approach is the *bisecting k-means* clustering approach, which has been shown to have good performance on document clustering [115].

Document clustering is traditionally viewed as a fully automated task without prior domain knowledge or user feedback, also often called the *unsupervised learning* method. However, in some cases, such as e-rulemaking, domain knowledge and user inputs are very desirable for improving the quality and usefulness the

clustering result.

In this work, we consider *domain knowledge* and *user feedbacks* during the clustering process, which is different from the traditional clustering approach. For *domain knowledge*, we consider the proposed rule and the identified three important aspects of the ERFR. For *user feedback*, we let users participate in the final selection of the issue and stakeholder terms. Both factors, which can be considered as constraints, are reflected in our novel *active feature selection* technique and *adaptive similarity measure*, which are the most important factors that affect clustering quality.

Recently, there have been some works on adding constraints to the document clustering process [124, 6, 61, 57]. Most of the constrains are specified at the instance level, i.e. the membership of documents. Reference [124] presents a variant of the *k-means* algorithm, in which users can specify what documents must be in the same cluster or in different clusters. Reference [61] also allows users to specify the documents' membership in the clustering process. In the undirected graph representation, each vertex represents a document and each edge indicates the similarity between two documents. The prior knowledge of documents' membership can be reflected by strengthening or weakening the corresponding edge. In [57], authors present a probabilistic model for clustering. In their approach, users can give their feedbacks to iteratively refine the clusters. Five types of user feedback are allowed in [57] when clustering emails, such as removing an activity cluster, specifying that an email belongs (or doe not belongs) to its assigned cluster, etc. Reference [6] also gives a probabilistic framework that integrates the distance-based and constraint-based approaches together.

**Document Summarization.** This is still a very active research topic. There are roughly three types of approaches for document summarization, namely (a) sentences-based (in which a summary consists of sentences extracted from the

original documents), (b) templates-based (in which predefined templates were filled as summarization), and (c) term-based (in which a set of carefully selected terms is used as cluster summary). Interested readers can refer to Section 2.3 for more detailed review of those approaches.

Our SCD is based on a selected set of terms, which is similar to the approach discussed in Chapter 4. But, the SCD can provide additional statistical information for the clusters than CD. In addition, we also consider situations when the clustering is hierarchical.

**E-Rulemaking.** This is a recently emerged research topic, and it has attracted a lot of attention from different research communities. In [67], authors try to analyze each document based on different aspects of text, such as argument structure, topics, and opinions. Their main focus was on the analysis of each feedback. They showed that such multidimensional text analysis could help highlighting the main focus of each feedback. Our work differs from theirs in many ways. First, our focus is not on each individual feedback but on the whole collection of feedbacks. We try to produce a high-level "map" so that it can be used as a "handle" to help the rule-writer to navigate the collection of feedbacks. Second, we perform clustering and construct SCDs at the same time, while the work in [67] does not. Third, we do not try to identify all opinion expressions in a feedback; instead, we are only interested in opinions that are related to those pre-identified issues and stakeholders.

In [120] and [17], authors try to produce succinct cluster labels, or cluster descriptions (CDs), for given document clusterings. Their results showed that such labeling efforts can help users better to understand and organize large collection of documents, such as e-rulemaking feedbacks. Our work also tries to produce succinct and informative labels for document clusters, but we are doing so at the same time when the clustering is performed. In addition, our SCD contains addi-

tional statistic information besides the terms, also considers the three important aspects of ERFR during selection process.

In this work, we also utilize some existing methodologies, such as F-score and *kappa*, for the evaluation of clustering quality and SCD quality.

## 5.3 Clustering of ERFR

When the vector-space-model is used, document clustering usually involves the following steps: First, a set of features (e.g., bag of words) is selected from the document corpus using some selection method. Second, each document is represented by a feature vector, which consists of weighting statistics of all features. Finally, clustering proceeds by measuring the similarity (e.g., a function of Euclidean distance) between documents and assigning documents to appropriate clusters [137].

In this work, we adopt the vector-space-model for representing documents. We also utilize the *bisecting k-means* algorithm for clustering the ERFR.

Below, we will introduce our novel feature selection method and our similarity measure strategy. We will also show how the clustering is performed based on the knowledge of opinions, issues and stakeholders.

### 5.3.1 Active Feature Selection

A good feature set should be able to discriminate dissimilar documents as much as possible and its dimensionality (namely the cardinality of the set) should be as low as possible. The TFxIDF (Term Frequency times Inverse Document Frequency) model [106], as discussed in Section 2.2, is one of the most popular models, and has been shown to be effective. In the TFxIDF model, terms that appear too rarely or too frequently are ranked lower than other terms that hold the balance.

Our feature selection approach, called *active feature selection (AFS)*, is mainly

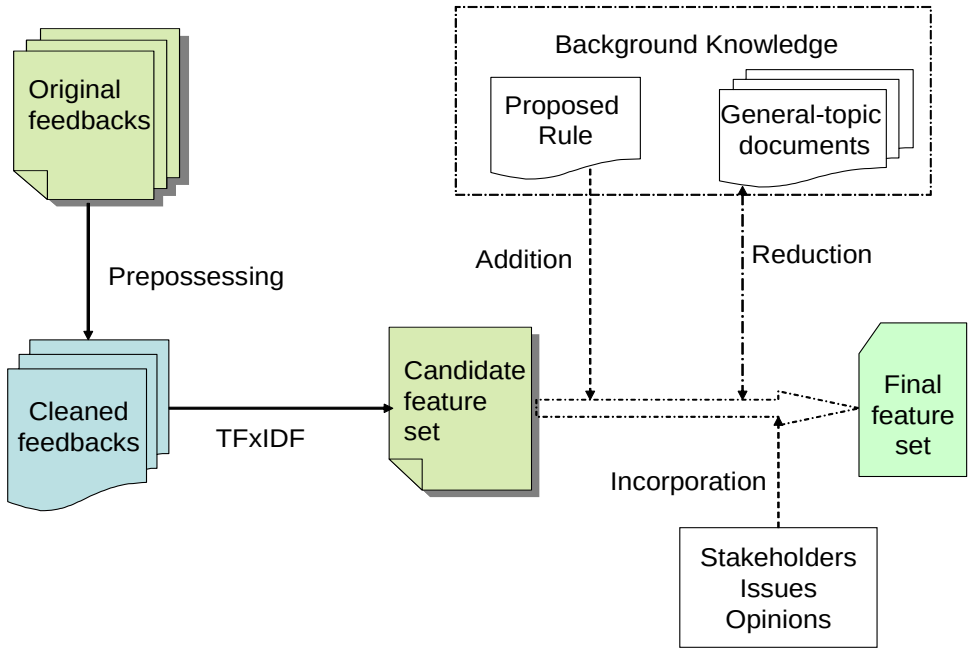



Figure 5.1: The Active Feature Selection (AFS) Process

based upon the TFxIDF model, but it has the following two additions. First, we consider the *background knowledge* when choosing features. Second, we incorporate all of the identified opinions, issues and stakeholders into the feature set.

Figure 5.1 shows the overall work flow for the AFS approach. We first perform the preprocessing steps (i.e. stopword removal, stemming, etc.) on the given ERFR. Then we calculate the TFxIDF value based on the "cleaned" feedbacks, and select candidate features based on the ranking of the TFxIDF values. We then consider the "background knowledge" and incorporate the O, I and S to obtain the final feature set.

Below, we will discuss how the background knowledge is considered and how the O, I and S are incorporated.

#### 5.3.1.1 Considering Background Knowledge

In the AFS approach, we not only consider those "target" feedbacks in which the clustering will be performed on, but also certain available *background knowledge*. We consider two types of background knowledge when dealing with the ERFR, the proposed rule and a collection of general-topic documents.

**The Proposed Rule.** In the "notice-and-comment" e-rulemaking process, the agencies first publish the proposed rule to the Federal Register, then they collect feedbacks from the interested parties. It is safe to say that most of the comments[2] directly comment on the proposed rules. Therefore, the proposed rule contains valuable information for organizing those feedbacks.

In this work, we utilize two types of information from the proposed rule for better feature selection. One is the *metadata* and the other one is the *frequent terms*. We will discuss each of them below.

By metadata, we refer to the information other than the actual rule description (or rule summary). For example, there are some metadata in the NoA-NOP rule summary, such as the rule name, "AGENCY", "ACTION", Docket Number, and etc. An example summary can be found in Appendix A.1.0. In this case, we add the agencies, which can be stakeholders, to our feature set, such as Agricultural Marketing Service (AMS) and USDA.

We also add some of the *frequent terms* to our feature set. We believe that terms that appear frequently in the proposed rule deserve more attention, even though they may not be frequent in the actual feedbacks. In this work, we choose the frequency threshold to be 5. Note that we count the frequency after the preprocessing steps, i.e. stop-word removal and stemming. Also note that those frequent terms added here can be pruned in the later steps. For example, based on the frequency count of the NoA-NOP rule summary, we add "organic", "program",

[2]There are examples that the feedbacks are not related to the proposed rule at all.

"product", etc. to the feature set. But, "product" will be pruned form the feature set because it happens to be a commonly used term for other document collections, which will be discussed next.

**General-topic Document Collection.** According to Zipf's law [92], some terms are more commonly used than others across all documents. In this work, we try to eliminate those terms that are commonly used in other general-topic document collections. Note that this approach takes a broader view by considering some other document collections, while the TFxIDF model only deals with the terms in the document collection under consideration.

Ideally we should gather a large collection of documents as *background knowledge*. Those documents should be collected from different source (such as newspapers, magazines, books and real life conversations), and cover variety of topics. Then, we can find out what terms are frequently used among those documents. Those frequent terms will be used as a "non-feature-list" in the feature selection process. Fortunately, there have been many works done on this by linguists. In this work, we obtained the 500 most commonly used words in the English language from http://www.world-english.org/english500.htm, we use them directly in our feature selection process. Note that, there are some overlaps between this list and the stopword list.

Formally, let $GT_\lambda$ be the $\lambda$ most frequent terms in the general-topic document collection, and let $FSC$ be the feature set candidates. The resulting feature set will have those frequent terms filtered out, i.e.

$$FSC_{new} = FSC \setminus GT_\lambda$$

In this work, we set $\lambda = 500$. By eliminating those commonly used terms, we can make the feature set more discriminative. More importantly, we can further reduce the dimensionality of the features, which is a big curse for document clustering [114].

As a side note, one can also consider domain-specific ontologies as background knowledge, and integrate them into the clustering process.

#### 5.3.1.2 Incorporating O, I and S

Usually the top $M$ terms with highest TFxIDF value will be selected as features, where $M$ is a user specified value (e.g. 1000). However, the terms that represent opinions (O), issues (I) and stakeholders (S) may not have very high TFxIDF value, therefore they may not be included in the feature set. Since those terms, as identified in Chapter 3, are important for organizing e-rulemaking feedback, we want to directly add them into the final feature set if they are not already in; that is, final feature set $FS = FSC \cup \{t \mid t$ is a selected opinion (O), issue (I) or stakeholder (S)$\}$.

### 5.3.2 Adaptive Similarity Measure

The TFxIDF values are calculated and normalized as follows:

$$w_{i,j} = tf_{i,j} * log_2(N/df_i) \tag{5.1}$$

$$W_{i,j} = \frac{w_{i,j}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}} \tag{5.2}$$

Here $w_{i,j}$ is the TFxIDF value of term $i$ in document $j$; $tf_{i,j}$ is the frequency of term $i$ in document $j$; $df_i$ is the number of documents where term $i$ occurs at least once; and $N$ is the total number of documents in the collection.

One important aspect of this work is that we are more interested in those opinion/issue/stakeholder terms when organizing ERFRs. We can emphasize the importance of those features by giving them more weights than that given by the normal TFxIDF model. For example, we can boost the weight of all O, I and S related features at once for flat clustering; or boost them separately to form

hierarchical clustering. We call the boosted weight values the *adaptive similarity measure* (ASM), since the boosting allows us to assign more importance to those special aspects.

We perform weight boosting in the following fashion. First, we boost the TFxIDF value of those OIS features, depending on the needs, to $w_{i,j} = w_{i,j} * (1 + \alpha)$, where $\alpha$ is a user defined parameter (e.g. $\alpha = 0.1$). The normalization (Formular 5.2) is then performed again after the boosting. Note that the TFxIDF value of those OIS features is not the real TFxIDF value; it is an amplified TFxIDF value. Therefore, we call them OIS-TFxIDF (OIS-weighted TFxIDF) values. We will show the impact of this $\alpha$ parameter in the experiment section.

In this work, each document is represented as a feature vector with the normalized OIS-TFxIDF values discussed above.

We use the cosine measure to compute the similarity between two documents. Given two document vectors $d_i$ and $d_j$, the cosine measure is defined by the cosine of the angle between the two vectors:

$$sim_{cos}(d_i, d_j) = \frac{d_i \cdot d_j}{\|d_i\| * \|d_j\|},$$

where $\cdot$ denotes the vector dot product and $\| \|$ denotes the length of a given vector. In this work, we use the simplified formula $sim_{cos}(d_i, d_j) = d_i \cdot d_j$, since the lengths of the feature vectors for all documents are the same.

### 5.3.3 Clustering Algorithm

The $k$-means algorithm and its variants have been widely used for document clustering due to its efficiency and implementation simplicity [68, 48]. Roughly speaking, the basic non-hierarchical $k$-means works as follows: First, it selects $k$ data points as the initial centroids, one for each of $k$ clusters. Second, each data point is assigned to the cluster whose centroid is the closest to the data point. Third,

the centroid of each cluster is recalculated. Steps two and three are repeated until the centroids do not change.

Reference [115] shows that the *bisecting k-means* outperforms basic $k$-means as well as agglomerative hierarchical clustering algorithms in terms of accuracy and efficiency. The bisecting $k$-means algorithm is shown in Algorithm 5.3.1. Initially, all data points are in one cluster. Iteratively, the algorithm selects a cluster to split, and then employs basic $k$-means to create two sub-clusters for that cluster; these two steps are repeated until the desired number $k$ of clusters is reached.

---

**Algorithm 5.3.1** The Bisecting k-means Algorithm

- *Initialization: put all the documents into a single cluster;*
- *While the desired number of clusters is not reached, do*
  1. *Pick a cluster (e.g. the largest) to split;*
  2. *Find* 2 *sub-clusters of that cluster using the* basic k-means *algorithm;*
- *Return the clustering.* ■

---

Note that step 2, the bisecting step, is usually repeated multiple times. The two sub-clusters will be chosen from the split so that the two clusters have the highest overall similarity. For each cluster, its similarity is the average pairwise document similarity; that is, for a given cluster $C_t$, the similarity is

$$sim_{cos}(C_t) = \frac{\sum_{d_i,d_j \in C_t; i<j} sim_{cos}(d_i, d_j)}{\|C_t\|}.$$

In this work, we utilize the bisecting $k$-means algorithm to form the clustering, either flat clustering or hierarchical. For flat clustering, the weights for all OIS features are boosted together. For hierarchical clustering, only one type of feature was used or only the weights of one selected type of feature was boosted. The bisecting $k$-means algorithm is then applied based on different order of O, I or S to form different clustering.

## 5.4 Constructing Succinct Cluster Description (SCD)

While the clustering is performed on ERFR, we also construct succinct cluster descriptions (SCD) at the same time. Our SCD is similar to the CDs discussed in Chapter 4, but with additional statistical information associated with it.

Below, we will first discuss what kind of statistical information will be added to the SCDs. Then, we will give a brief review of the CDD measure. We also discuss another surrogate measure, called the CU measure, which is based on the category utility measure [10, 33, 38]. After that, we will introduce the enhanced PagodaCD algorithm, called PagodaCD+, to construct the SCD by using either the CDD or CU measure.

### 5.4.1 Additional Statistical Information for SCD

In addition to the terms in the SCD, we also gather some statistical information about those terms so that to give users more quantitative information. We consider such information for each individual term and all the terms as a whole. Note that users have the options on whether and how to display such information.

- For each individual term in the SCD, we list its document frequency within each cluster and in the collection. We also mark the terms if they are opinions, issues or stakeholders. For example, *landowner(S,128/213)* means *landowner* appeared in 128 documents within the cluster and in 213 documents in the collection; and it happens to be a stakeholder term.

- For all the terms as a whole, we list the number of documents covered by those terms within the cluster and in the collection. We can also display the values of the three factors: *coverage*, *disjointness* and *diversity*.

- For each term in the SCD, we list the posterior probability for the term

to occur in the given cluster; for a given cluster $C_t$, a term $t_i$ occurs in it with the probability of $P(t_i|C_t)$. Based on the Bayesian probability and the multinomial model, the probability with Laplacian smoothing can be obtained by equation 5.3, in a way similar to the one used in [78]:

$$P(t_i|C_t) = \frac{1 + \sum_{j=1}^{|D|} tf_{(i,j)} P(C_t|d_j)}{|T| + \sum_{s=1}^{|T|} \sum_{j=1}^{|D|} tf_{(s,j)} P(C_t|d_j)} \tag{5.3}$$

Here, $|D|$ is total number of documents, $|T|$ is total number of terms, $tf_{(i,j)}$ is the frequency of term $t_i$ in document $d_j$, and $P(C_t|d_j) \in \{0, 1\}$ is the probability that document $d_j$ belongs to cluster $C_t$ (which depends on the terms of the documents in the cluster).

■

### 5.4.2 The CDD Measure

In Section 4.5.2, we defined the CDD measure for efficient search of high quality CDs. Recall that the CDD measure is defined in terms of the three factors of *coverage*, *disjointness* between terms across CDs for different clusters, and *diversity* among terms within the CD of one cluster. Notice that while the *disjointness* is defined on CDs for multiple clusters in one clustering, the other two only involve CDs for single clusters.

Briefly, for the old CDs $CD_1^o, ..., CD_L^o$ and new CDs $CD_1^n, ..., CD_L^n$, the *CDD improvement* is defined by

$$\Delta\mathsf{CDD} = \begin{cases} \delta(\mathsf{Cov}) + \delta(\mathsf{Dis}) + \delta(\mathsf{Div}), & \text{if } min(\delta(\mathsf{Cov}), \delta(\mathsf{Dis}), \delta(\mathsf{Div})) \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

The *improvement* of the factors is defined as:

$$\delta(\mathsf{Cov}) = \frac{\mathsf{Cov}_{C_j}(CD_j^n) - \mathsf{Cov}_{C_j}(CD_j^o)}{\mathsf{Cov}_{C_j}(CD_j^o)},$$

$$\delta(\mathsf{Dis}) = \frac{\mathsf{Dis}(CD_1^n, ..., CD_L^n) - \mathsf{Dis}(CD_1^o, ..., CD_L^o)}{\mathsf{Dis}(CD_1^o, ..., CD_L^o)},$$

$$\delta(\mathsf{Div}) = \frac{\mathsf{Div}_{C_j}(CD_j^n) - \mathsf{Div}_{C_j}(CD_j^o)}{\mathsf{Div}_{C_j}(CD_j^o)}.$$

Note that we insist that each improvement is non-negative. Also experiment results suggest that the "sum" of the individual improvements performs better than the "product".

### 5.4.3 The CU Measure

We now introduce the CU measure, which is another surrogate measure for constructing SCDs. The *category utility* measure was introduced in [10, 33, 38] for conceptual clustering. We will also use the improvement of this measure in our search process.

Category (which should be viewed as a synonym of cluster) utility is a tradeoff between intra-category similarity and inter-category similarity. For each term $t$ and cluster $C_i$, let $P(t)$ be the probability of $t$ in $\cup_{i=1}^{L} C_i$, $P(C_i|t)$ the probability of $C_i$ given $t$, and $P(t|C_i)$ the probability of $t$ in $C_i$. The *category utility* of $CD_1, ..., CD_L$ for a clustering $C_1, ..., C_L$ is given by

$$CU(CD_1, ..., CD_L) = \Sigma_{i=1}^{L} \Sigma_{t \in \cup_{j=1}^{L} CD_j} P(t) P(C_i|t) P(t|C_i).$$

By the Bayes rule, we also have

$$CU(CD_1, ..., CD_L) = \Sigma_{i=1}^{L} P(C_i) \Sigma_{t \in \cup_{j=1}^{L} CD_j} P(t|C_i)^2.$$

The category utility measure can be directly used as a surrogate measure. For the old and new CDs, $CD_1^o, ..., CD_L^o$ and $CD_1^n, ..., CD_L^n$, the *CU improvement* is defined by

$$\Delta\mathsf{CU} = \frac{(\mathsf{CU}(CD_1^n, ..., CD_L^n) - \mathsf{CU}(CD_1^o, ..., CD_L^o))}{\mathsf{CU}(CD_1^o, ..., CD_L^o)}.$$

■

### 5.4.4 Algorithm for Constructing SCD: Pagoda+

We now introduce the PagodaCD+ algorithm for constructing SCDs. The PagodaCD+ algorithm is an enhanced version of the PagodaCD algorithm, which was discussed in Section 4.6.

Roughly speaking, the PagodaCD+ algorithm has the following two enhancements over its predecessor PagodaCD . First, while selecting the SCD terms, it also calculates and records the desired statistical information associated with the terms. Second, it is made more general so that it can use any quality measurement, such as CDD measure, CU measure and the F-score . In addition, it also takes into consideration the three important aspects of ERFR (i.e. O, I and S) when selecting the candidate terms.

---

**Algorithm 5.4.1** The PagodaCD+ Algorithm

Inputs*: Clusters $C_1, ..., C_L$; $k$ (SCD size); $baseSize$; $incSize$; $minImp$; measure $M$;*

Outputs*: SCDs ($SCD_1, ..., SCD_L$) for the clusters*

Method*:*

1. *For each $i$ ($1 \leq i \leq L$), set $SCD_i$ to $\emptyset$, and let $CP_i$ consist of the candidate terms for cluster $C_i$;*

2. IterReplaceM*($\overline{CP}$, $\overline{SCD}$, $minImp$, $baseSize$, $M$);*
   *//$\overline{CP}$ and $\overline{SCD}$ are vectors of CPs and SCDs*

3. *For $j = 1$ to $\frac{k-baseSize}{incSize}$* IterReplaceM*($\overline{CP}$, $\overline{SCD}$, $minImp$, $incSize$, $M$);*

4. *For each $i$, calculate and gather statistical information for $SCD_i$ and each term in $SCD_i$;*

5. *Return SCDs $(SCD_1, ..., SCD_L)$.* ■

---

Parameter $k$ is the number of terms desired in each SCD, $baseSize$ is the number of terms to be obtained for each $SCD_i$ in the first major step, and $incSize$ is the number of terms to be added for each $SCD_i$ in each subsequent major step. Parameter $minImp$ is a user given minimum quality improvement threshold. $M$ is the given measure for searching SCDs (e.g. CDD , CU , F-score or others).

Notice that $CP_i$ is a pool of candidate terms for cluster $C_i$. The candidate terms can be chosen in different ways. In Section 4.6.3, we introduced two approaches: one is to choose from some of the most frequent terms of each cluster (e.g. 50); another one is the *DFD-based preselection* approach. Those approaches can still be applied here. However, users can add the OIS terms occurring in cluster $C_i$ to the candidate pool based on the needs.

The IterReplaceM procedure works in a way similar to the IterReplace discussed in Section 4.6.2, except that it has an additional parameter to indicate the measure used in search. From procedure 5.4.1 we can see that IterReplaceM is used to select $stepSize$ new terms for each $SCD_i$, while keeping the terms selected in previous levels unchanged. It first selects the most *prominent* (in terms of feature weighting scheme) $stepSize$ unused terms from the candidate term pools, and then use the given $M$ measure to repeatedly select the best replacement terms. For each iteration, it finds the best replacement term among all clusters and all terms for the current major step. This is repeated until no replacement term with significant quality improvement is found.

---

**Procedure 5.4.1** IterReplaceM($\overline{CP}$, $\overline{SCD}$, $minImp$, $stepSize$, $M$)

*// Vectors $\overline{CP}$ and $\overline{SCD}$ are passed by reference;*

1. *For each $i$ ($1 \leq i \leq L$), let $CDTL_i = \{$the most prominent stepSize of terms in $CP_i - SCD_i\}$, and let $SCD_i = SCD_i \cup CDTL_i$;*

2. *Repeat until no replacement is found:*

   - *For each $i$, old term $t^o \in CDTL_i$ and new term $t^n \in CP_i - SCD_i$, compute $\Delta M_i(t^o, t^n)$ for the hypothetical replacement of $t^o$ in $SCD_i$ by $t^n$. Suppose the best replacement among all possible ($t^o$,$t^n$) pairs for $i$ is $\Delta M_i(t_i^o, t_i^n)$, achieved at $t_i^o$ and $t_i^n$.*

   - *Let $C_j$ be the cluster with the largest $\Delta M_i$, i.e.*
     $\Delta M_j(t_j^o, t_j^n) = \max_i \Delta M_i(t_i^o, t_i^n)$.
     *If $\Delta M_j(t_j^o, t_j^n) > minImp$, then*
     *(a) let $SCD_j^n$ (respectively $CDTL_j$) be the result of replacing $t_j^o$ in $SCD_j$ (respectively $CDTL_j$) with $t_j^n$,*
     *(b) replace $SCD_j$ by $SCD_j^n$, and*
     *(c) keep the other $SCD_i$ unchanged.* ■

---

Notice that PagodaCD+ uses IterReplaceM to do the replacement only in a local one-layer-at-a-time manner. This leads to both faster computation and the monotone-quality behavior (getting higher F-score when CDs become larger). The computation complexity is similar to the analysis on the PagodaCD. The only extra cost is the calculation of the statistical information for those SCD terms, which is not significant.

## 5.5 Experimental Evaluation

Evaluating the qualities of the produced clustering and SCDs is a non-trivial task, the ultimate test being user judgment. However, human-centered evaluation will be subjective, time-consuming, expensive, and perhaps inconsistent. In this work,

we mainly use alternative objective measures (e.g. F-score), and use limited human efforts at small scales.

In the following subsections, we will first briefly introduce our experiment setup. Then, we will give a small example to illustrate the idea of our OIS-based approach. Finally, we will present some experimental results to show the quality of our clustering and SCD based on some real-world data sets.

### 5.5.1 Experiment Setup

**Data sets.** In this experiment, we will use several publicly available e-rulemaking data sets, which were briefly descried in Section 2.1.4. Since the *DoA-NOP* collection contains more feedbacks, and also seems more "original" than the other four collections that we will consider, therefore, the majority of our experiments will use this data set. We constructed several subsets (i.e. *d5-1k*, *d5-2k*, *d5-3k* and *d5-4k*) from the volume-1 distribution of the *DoA-NOP* collection. They were constructed in the following manner: we obtained 22 folders after unpacking the volume-1. The folders are numbered, and each folder has several hundreds of feedbacks. We then eliminated those feedbacks contain little or no meaningful textual content, such as "Test only", "test test", etc. Finally, we chose certain desired number of feedbacks starting from the beginning (smallest index first); e.g. *d5-1k* contains the first 1000 feedbacks, *d5-2k* contains the first 2000 feedbacks, and so on. In addition, we also randomly selected 50 feedbacks from *d5-1k*, called *d5-50*. This small data set was used to manually verify the quality of our approaches, since the smaller size makes it possible for us to read those feedbacks and manually annotate them according to our understanding of those feedbacks.

We also use the *Reuter2k* data set to test the clustering quality, since documents in *Reuters-21578* have been annotated by human and pre-classified; construction of the *Reuter2k* was described in Section 4.8.2.

Table 5.1 is a summary of the data sets used in the experiments.

| **Data sets** | **Source** | **# of docs** | **Is ERFR** |
|---|---|---|---|
| d1 | EPA-CWA | 500 | Y |
| d2 | EPA-NESHAP | 2000 | Y |
| d3 | DoT-CAFE | 990 | Y |
| d4 | DoA-SWPM | 955 | Y |
| d5-50 | DoA-NOP | 50 | Y |
| d5-1k | DoA-NOP | 1000 | Y |
| d5-2k | DoA-NOP | 2000 | Y |
| d5-3k | DoA-NOP | 3000 | Y |
| d5-4k | DoA-NOP | 4000 | Y |
| Reuter2k | Reuter-21578 | 2000 | N |

Table 5.1: Summary of the Data Sets

**Environment.** All experiments were conducted on a PC with 2.8GHZ CPU and 1GB RAM, running Windows XP. All programs were coded in *Java*.

### 5.5.2 An Example Illustration

Before we get into the results for large data sets, let us look at a small example that illustrates the OIS-based approach to perform e-rulemaking feedbacks clustering and to construct SCDs for each cluster.

Assume that we deal with the feedbacks about revising the Clean Water Act (CWA)[3] proposed by the Environmental Protection Agency (EPA) in 2002. For simplicity, also assume that three important issues as conceived by the public are *the definition of "water of United States" (noted as DWUS)*, *navigation*, and *pol-*

[3]The feedback fragments used in this example are from actual feedbacks.

*lution*; three major stakeholders are *landowners*, *children*, and *environmentalists*; three types of opinions are considered: *in favor of*, *against* and *others* (new idea or don't know or don't care). In the OIS-based approach, rule writers and analysts have the flexibility to choose how the clustering will be performed (i.e. flat or hierarchical), and what kind of statistical information will be displayed.

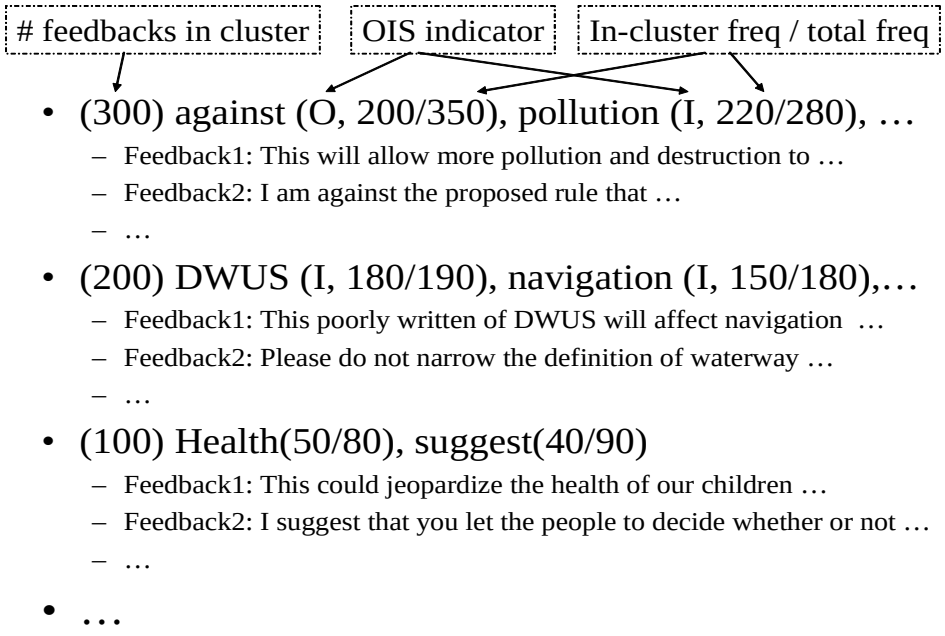



Figure 5.2: Flat Clustering with SCDs

**Flat Clustering.** In this case, our approach is similar to the traditional content based approach. But, we take consideration of the O, I and S terms throughout the process. In some cases, the impact of O, I and S can be significant on the results of clustering and SCDs. For example, a majority of the feedbacks with comments on *pollution* could be put into same cluster, or feedbacks that *in favor of* the rule will be in the same cluster. In addition, if those OIS related terms can be selected as SCD terms, the overall theme of the underling feedbacks will be highlighted dramatically. For example, the terms of *pollution* and *against* were selected as as

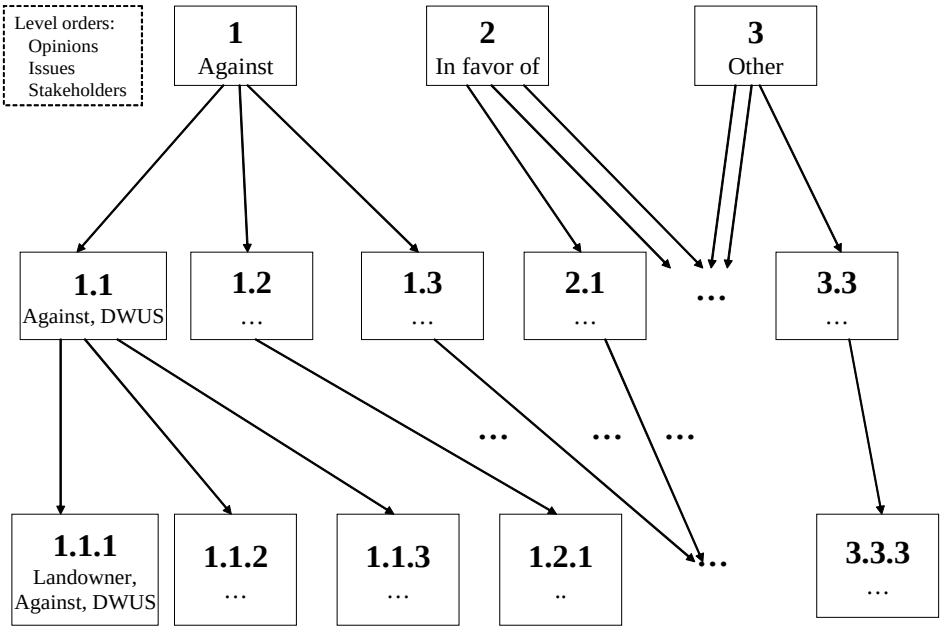



Figure 5.3: Hieratical Clustering with SCDs in the Order of O-I-S

SCD terms in Figure 5.2. Therefore, the clustering, along with the SCDs can be a great helper for the rule-writers to digest those feedbacks efficiently.

**Hierarchical Clustering.** In this case our approach can take consideration of O, I and S in different order (and maybe different weights) so that different clusterings can be generated. For example, the clustering can be performed based on the order of O-I-S, as illustrated in Figure 5.3. In this clustering, the top level was clustered with main emphasis on the *opinion* aspect; therefore, three clusters were formed, i.e. "in favor of", "against" and "other". The middle level was based on *issues*; hence, three clusters were generated for each cluster in the top level, i.e. "DWUS", "navigation", "pollution". The bottom level was based on *stakeholder*; again, three clusters were formed for each cluster in the middle level, i.e. "landowners", "children", "environmentalists".

The SCDs become even more attractive in the case of hierarchical cluster-

ing. The top level SCD will address the aspect of *opinions*, which will include all the feedbacks. For example, cluster 1 may be labeled as "against". The middle level adds more details about the *issues* that support the top level orientation ("against"). For example, cluster 1.1 may have labels of "DWUS" and "against". The bottom level concerns about the *stakeholders*. For example, cluster 1.1.1 may have labels of "landowners", "DWUS" and "against". Those carefully selected CD terms in each level, with reasonable interpretation, could be very descriptive "handles" to the rule-writers to manage and digest the ERFR. Additional statistical information of SCDs gives some sense about the confidence of those SCDs.

We can see that the clustering structure, along with the SCDs, is an excellent navigation aid for the rule-writers and analysts to organize and digest the ERFR efficiently. By utilizing the aid, they can easily drill down to certain cluster of the feedbacks to look at the details.

Other combinations, such as I-S-O or S-I-O, can also be used. Note that a user can also combine two aspects into one, in which two aspects will be considered together in the same clustering level, for example O-(I,S).

### 5.5.3 Evaluating Clustering Quality

We consider that a clustering is good if *similar* feedbacks were grouped into same clusters. Traditional content-based clustering approaches consider the similarity purely based on the co-occurrence of some terms. In our OIS-based approach, the similarity is also determined by how coherent those feedbacks are based on those three important aspects of ERFR. In some sense, it takes consideration of the conceptual similarity, not just the lexical similarity.

Below, we will first describe our evaluation methodology. Then we will show some experimental results. Here, we mainly focus on the clustering quality, and we will examine the SCD quality next.

#### 5.5.3.1 Evaluation Methodology

Besides some small-scale human judgment, we mainly use the F-score to measure the quality of clusterings, which has been widely used by other researchers.

Assume the *supposed-to-be* clustering is $\mathcal{C} = \{C_1, ..., C_L\}$, and the *algorithm-generated* clustering is $\mathcal{C}' = \{C'_1, ..., C'_L\}$. The overall difference between the algorithm-generated clustering and the suppose-to-be clustering is defined as the weighted average of the F-score of the component clusters:

$$F(\mathcal{C}', \mathcal{C}) = \sum_{i=1}^{L} \frac{|C_i|}{|D|} \max_{C_j \in \mathcal{C}} F(C'_i, C_j),$$

where $D = \cup_{i=1}^{L} C_i$. $F(C'_i, C_i)$ denotes the F-score for $C'_i$ and $C_i$, and is defined by $F(C'_i, C_i) = \frac{2 * P(C'_i, C_i) * R(C'_i, C_i)}{P(C'_i, C_i) + R(C'_i, C_i)}$, where $P(C'_i, C_i) = |C'_i \cap C_i|/|C_i|$ is the *precision* and $R(C'_i, C_i) = |C'_i \cap C_i|/|C'_i|$ is the *recall*.

Note that for a hierarchical clustering, the F-score of $\max_{C_j \in \mathcal{C}} F(C'_i, C_j)$ is the maximum value it attains at any node in the hierarchical tree. That is, $\mathcal{C}$ will be the hierarchical tree in this context.

#### 5.5.3.2 Experimental Results

For the e-rulemaking feedbacks, we do not have the *supposed-to-be* clustering to begin with. To manually annotate each feedback involves massive human efforts, which is infeasible for large collections. In this work, we use the *d5-50* and *Reuter2k* data sets to evaluate our clustering approach, since we have manually classified the feedbacks in *d5-50* to three clusters, and each article in *Reuter2k* has also been pre-classified to one of the eight clusters.

Note that with those labeled data sets, we can use F-score to judge the quality of the clustering approach, like other researchers do. However, because human labeling is subjective, and sometimes is impossible, F-score alone may not reflect the true quality of the clustering. Therefore, instead of solely focusing on the F-score,

we will also pay attention to the techniques that affect the F-score. Sometimes the changes make sense for humans to understand the clustering, but may not help to improve the F-score.

Below we will mainly evaluate the effectiveness of our *active feature selection (AFS)* and *adaptive similarity measure (ASM)* techniques used in the OIS-based clustering approach. Note that we consider the normal Vector-Space-Model with cosine similarity as our baseline. The clustering was also performed by using the Cluto implementation of the bi-secting k-means.

| **Data sets** | **# of Clusters** | **Baseline F-score** |
|---|---|---|
| d5-50 | 3 | 0.6245 |
| Reuter2k | 8 | 0.5956 |

Table 5.2: Clustering Results in F-Score for Baseline Settings

Table 5.2 shows the baseline clustering results for the data sets *d5-50* and *Reuter2k*. Tables 5.3 and 5.4 show some similarity statistics of those clusterings. As introduced in Section 4.8.3, *CID* is the cluster id, *Size* the number of documents in a cluster, *ISim* the internal similarity, *ESim* the external similarity, and *ISdev* and *ESdev* are the standard deviation of the internal and external similarities, respectively. Note that the clustering process tries to assign smaller CIDs to those clusters that have higher inter-cluster similarity.

| **CID** | **Size** | **ISim** | **ISdev** | **ESim** | **ESdev** |
|---|---|---|---|---|---|
| 0 | 5 | 0.502 | 0.140 | 0.016 | 0.004 |
| 1 | 12 | 0.126 | 0.016 | 0.029 | 0.013 |
| 2 | 33 | 0.087 | 0.015 | 0.027 | 0.010 |

Table 5.3: Internal and External Similarities for 3-ways Clustering in *d5-50*.

| CID | Size | ISim | ISdev | ESim | ESdev |
|---|---|---|---|---|---|
| 0 | 90 | 0.384 | 0.103 | 0.061 | 0.019 |
| 1 | 71 | 0.302 | 0.092 | 0.025 | 0.015 |
| 2 | 216 | 0.310 | 0.110 | 0.050 | 0.016 |
| 3 | 486 | 0.203 | 0.073 | 0.039 | 0.013 |
| 4 | 123 | 0.111 | 0.045 | 0.012 | 0.004 |
| 5 | 266 | 0.058 | 0.021 | 0.013 | 0.008 |
| 6 | 370 | 0.041 | 0.015 | 0.015 | 0.007 |
| 7 | 378 | 0.030 | 0.009 | 0.016 | 0.009 |

Table 5.4: Internal and External Similarities for 8-ways Clustering in *Reuter2k*.

**Impact of the AFS Approach.** The AFS approach not only considers the target documents, but also the *background information*. For e-rulemaking data sets, this approach also considers the proposed rule itself and the OIS terms that identified for the collection.

For the *d5-50* data set, filtering out commonly used words does not change the clustering results. This is simply because the data set is too small to be impacted by this filtering. However, it has obvious impact on the *Reuter2k* data set. The F-score of the clustering is improved from 0.5956 to 0.615. But, the improvement of SCD was significant, which will be seen later in Table 5.9.

When we consider the proposed rule and OIS terms for *d5-50*, the clustering results improved a little bit. We will discuss the impact in conjunction with the ASM approach next.

**Impact of the ASM Approach.** The ASM approach can amplify the weight of OIS terms based on user's needs. Table 5.5 shows the clustering results in F-score when no adjustment was made (baseline) and when the adjustment were made based on different strength, $\alpha$. Note that the $\alpha$ is defined in Section 5.3.2 to

| Baseline (No Boost) | Boot OIS Together | | |
|---|---|---|---|
| | $\alpha = 0.1$ | $\alpha = 0.2$ | $\alpha = 0.3$ |
| 0.6245 | 0.643 | 0.645 | 0.645 |

Table 5.5: Impact of the ASM Technique in Terms of F-score for Data Set *d5-50* When Number of Clusters is 3

calculate the new OIS-TFxIDF value.

We can see that the ASM weight adjustment, i.e., OIS-based clustering approach, improves the clustering quality. However, the results do not change anymore when $\alpha$ is at certain point, 0.3 in this case. This results suggest that O, I and S are important aspects for manage e-rulemaking feedbacks, since users tend to group feedbacks based on those factors. However, because there are only limited number of terms that are considered as O, I and S, the impact will be saturated at certain point. Note that it is also possible that the clustering results will deteriorate if we give too much emphasis on those OIS terms (in extreme case solely rely on them). In later experiments, we set $\alpha = 0.1$, unless specified otherwise.

**Flat vs. Hierarchical Clustering.** Hierarchical clustering can be obtained by recursively applying the flat clustering algorithm. The clustering quality at a particular node of the hierarchy can be measured by weighted average F-score of its immediate children. However, it is hard to get an objective value, since data sets usually are not labeled hierarchically. Especially, evaluation becomes more difficult for those data sets with few labeled classes. For example, the *d5-50* only has three clusters. Even if we only perform 2-way clustering twice, we will still end up with four clusters. Therefore, it is hard to get a meaningful F-score.

For the *Reuter2k* data set, it is possible to get meaningful F-score hierarchically, since there are 8 pre-defined classes. Figure 5.4 shows the clustering results for *Reuter2k* with 3-level hierarchical clustering, in which 2-way clustering were per-

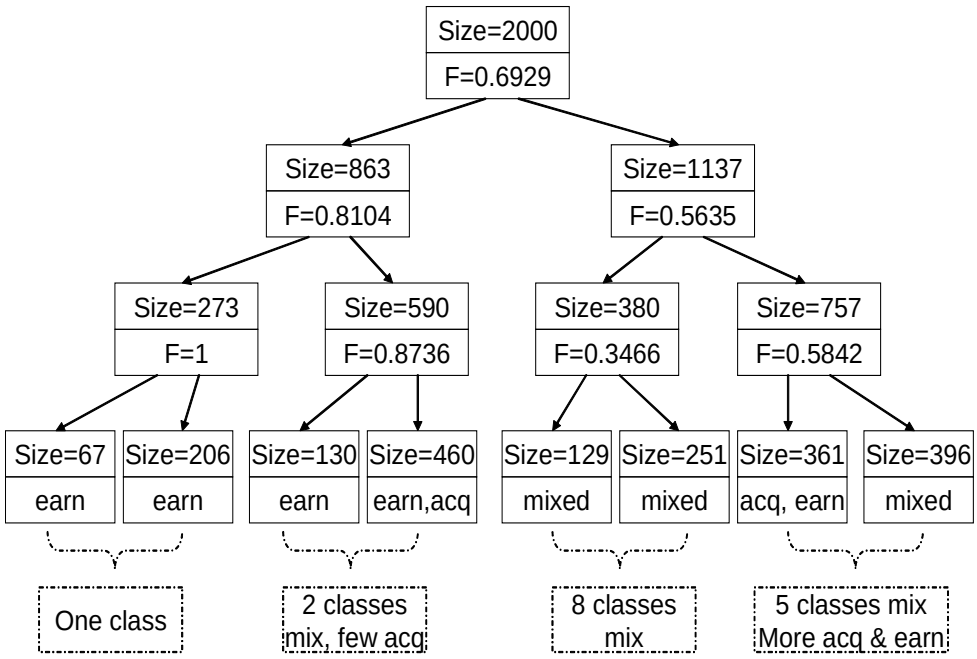



Figure 5.4: Hieratical Clustering Results for *Reuter2k*

formed at each level. The figure shows the number of documents for each cluster and the F-score obtained at each non-leaf node.

In this work, we are more interested in the hierarchical structure for those e-rulemaking data sets, especially when the levels were based on different order of O, I and S. Therefore, instead of checking for the F-score, we will pay more attention to the content of the clustering, that is, to check if the clustering helps users to digest the underlying documents.

Figure 5.5 illustrates the document distribution when 2-way clustering performed on *d5-50* for 3 hierarchical levels. The aspect of O, I and S was emphasized, one at each level respectively, when the clustering was performed. We can also see some example feedbacks for the leaf node. Note that the clustering can also be performed using different OIS order, which could impact the clustering quality. In this example, we observed that the order of OIS produces the best result.

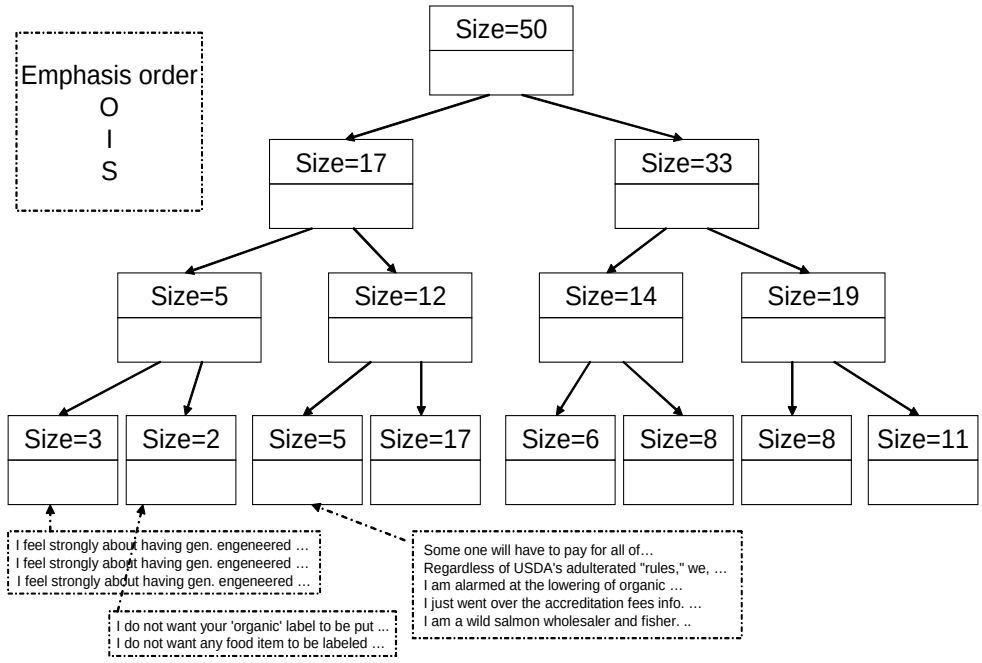



Figure 5.5: Hieratical Clustering Result for *d5-50* in the Order of O-I-S

As said before, the clustering is only one part of the SD. Next, we will see the OIS-based approach also produce high quality SCDs.

### 5.5.4 Evaluating SCD Quality

We now turn to the evaluation of SCDs' quality. Recall that the purpose of SCDs is to provide an informative "handle" to the users so that they can quickly get a high-level sense of what the clusters contain. Therefore, the quality will be largely determined by how well the SCDs can help the users to "visualize" the underling content. Note that users can be human beings or roboticized programs.

In the following subsections, we will first introduce our methods to evaluate the SCD quality, including the selected terms and the associated statistics. Then we will show our experimental results on some data sets.

#### 5.5.4.1 Evaluation Methodology

While pure human judgement is necessary and is the ultimate measure, we mainly use the *CD-based classification* approach to evaluate the quality of SCDs.

The main idea of the *CD-based classification* approach, as introduced in Section 4.4.1, is to use the terms in SCDs to build classifiers that can be used to simulate the process of human interpretation/visaluzation of the SCDs. The classification results can be viewed as interpreted clusters that contain what users believe are in the clusters. The amount of difference between the interpreted and the original clusters can then measure the quality of the CDs.

Let the interpreted clusters be $\mathcal{C}'' = \{C''_1, ..., C''_L\}$ , and the algorithm-generated clustering be $\mathcal{C}' = \{C'_1, ..., C'_L\}$. The CD quality would then be measured using the F-score between $\mathcal{C}''$ and $\mathcal{C}'$ , denoted as $F(\mathcal{C}'', \mathcal{C}')$, which has been defined in Section 5.5.3.1.

The statistical information of each term in the SCD can indicate its weight, i.e. importance. The score for the SCD as a whole can show the effectiveness of those terms collectively as a "handle" to the cluster.

| **Approaches** | **Note** |
|---|---|
| Descriptive CD | Terms with higher TFxIDF values in the centroid vector |
| Discriminating CD | Terms with higher $\frac{thisclusterfrequency}{otherclustersfrequency}$ |
| Frequency-based CD | Most frequent terms for each cluster |
| LN-Search CD(CU) | Local-neighbor search using CU measure |
| CumulativeCD | Using CDD measure with DFD candidate pool |
| PagodaCD+(CDD) | Pagoda+ using CDD measure |
| PagodaCD+(CU) | Pagoda+ using CU measure |

Table 5.6: SCD Approaches Considered

#### 5.5.4.2 Experimental Results

Table 5.6 summarizes the approaches[4] that we will evaluate in this experiment. We will pay more attention to two measures (i.e. CDD and CU) and two algorithms (i.e. PagodaCD+ and *CumulativeCD*). Note that all SCDs are constructed at the same time when the clustering is performed. Also, unless indicated otherwise, all data sets are divided into 4 clusters at each level.

Figure 5.6 shows the SCD quality of different approaches in terms of F-score for e-rulemaking data sets *d1, d2, d3* and *d4*. We can see that the *Cumulative CD* and PagodaCD+ (both CDD and CU measures) approaches perform much better than other approaches. Note that for the local-neighbor search approach, we set NUMLOCAL = 50 and MAXNEIGHBOR = 20.

Figure 5.7 shows the results for the *Reuter2k* data sets. Again, the results show that the *Cumulative CD* and PagodaCD+ (both CDD and CU measures) outperform other approaches for non-ERFR data set.

**CDD vs. CU Measures.** From the above results, we can see that both measures performed well. Even though CU did better than CDD in some cases, overall, the CDD measure is better than CU. However, one advantage about the CU is that it has less parameters that need to be tuned than the CDD measure.

**Effectiveness of Different Search Algorithms.** In general, PagodaCD+ (both CDD and CU ) produces better SCDs than other approaches because of the layer-based nature and because it considers all the clusters when searching. SCDs produced by PagodaCD+ also have the monotone quality behavior, giving higher quality SCDs when more terms are in the SCDs. Because of the effectiveness of the CDD measure, the *CumulativeCD* approach also performs well in most of the cases, even though it considers one cluster at a time. One advantage of the *CumulativeCD* approach over PagodaCD+ approaches is that it uses less computation

[4]The Cluto package can also produce Descriptive and Discriminating CDs.

| **Data set: D1** | | | | |
|---|---|---|---|---|
| | **SCD Size** | | | |
| | **4** | **8** | **12** | **16** |
| Descriptive CD | 0.619806 | 0.340741 | 0.291048 | 0.318637 |
| Discriminating CD | 0.608442 | 0.616142 | 0.58207 | 0.569072 |
| Frequency-based CD | 0.338636 | 0.352232 | 0.314848 | 0.304697 |
| LN-Search CD (CU) | 0.67043 | 0.681613 | 0.649619 | 0.699172 |
| Cumulative CD | 0.685437 | 0.7049 | **0.746885** | 0.775156 |
| PagodaCD+ (CDD) | **0.715074** | **0.75641** | 0.739673 | **0.789079** |
| PagodaCD+ (CU) | 0.643167 | 0.695557 | 0.710137 | 0.762581 |
| | | | | |
| **Data set: D2** | | | | |
| Descriptive CD | 0.543082 | 0.486356 | 0.489794 | 0.477739 |
| Discriminating CD | 0.616991 | 0.569872 | 0.569112 | 0.570649 |
| Frequency-based CD | 0.279169 | 0.260219 | 0.259299 | 0.471089 |
| LN-Search CD (CU) | 0.696283 | 0.933362 | 0.847404 | 0.889278 |
| Cumulative CD | **0.987461** | **0.98696** | **0.986458** | **0.987467** |
| PagodaCD+ (CDD) | 0.940937 | 0.942485 | **0.984954** | **0.983948** |
| PagodaCD+ (CU) | 0.689086 | 0.69622 | 0.692061 | 0.687228 |
| | | | | |
| **Data set: D3** | | | | |
| Descriptive CD | 0.486207 | 0.480581 | 0.449098 | 0.450342 |
| Discriminating CD | 0.48611 | 0.612491 | 0.605721 | 0.596174 |
| Frequency-based CD | 0.160727 | 0.160607 | 0.160607 | 0.160607 |
| LN-Search CD (CU) | 0.648152 | 0.731824 | 0.721531 | 0.712923 |
| Cumulative CD | **0.741933** | 0.738803 | 0.737228 | 0.735646 |
| PagodaCD+ (CDD) | 0.71556 | 0.730974 | 0.733448 | 0.729972 |
| PagodaCD+ (CU) | **0.744271** | **0.751873** | **0.750408** | **0.748385** |
| | | | | |
| **Data set: D4** | | | | |
| Descriptive CD | 0.958038 | 0.937727 | 0.932992 | 0.929047 |
| Discriminating CD | 0.967141 | 0.956515 | 0.949423 | 0.949423 |
| Frequency-based CD | 0.941277 | 0.934768 | 0.912927 | 0.912927 |
| LN-Search CD (CU) | 0.969279 | 0.961086 | 0.946918 | 0.93565 |
| Cumulative CD | **0.977833** | **0.973491** | **0.969047** | **0.960073** |
| PagodaCD+ (CDD) | **0.979626** | **0.973378** | **0.964824** | **0.962876** |
| PagodaCD+ (CU) | 0.969339 | 0.969648 | **0.965674** | **0.960096** |

Figure 5.6: SCD quality in terms of F-Score vs. SCD Size (4, 8, 12 and 16) for Data Set *d1, d2, d3* and *d4*

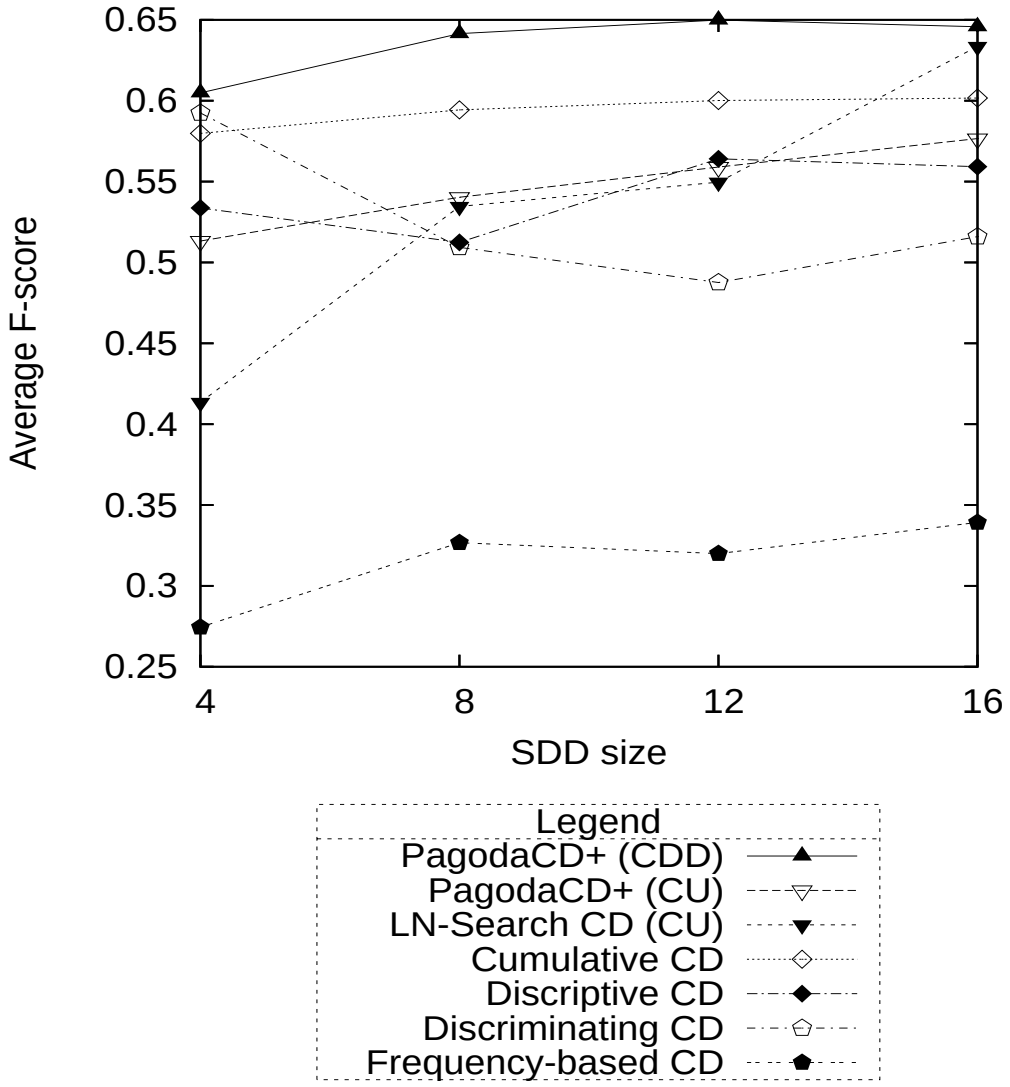



Figure 5.7: SCD quality in terms of F-Score vs. SCD Size for *Reuter2k*

time.

**CD Terms and Additional Information.** The SCDs not only contain the terms, but also additional information about the terms. Figure 5.8 shows the SCDs produced by different approaches for the data set *d5-1k* with SCD size set to 4. After each term, there is in-cluster document frequency and total document frequency information. While we prefer terms that appear in more documents, we also pay more attention to the ratio of in-cluster frequency over total frequency. For each term in the SCD, there is also the probability information defined by formula 5.3. Usually a larger value indicates a better choice. In addition, there is also the OIS indicator if it is applicable. Note that all terms are in their root format. As an alternative, one can also replace them to their non-root form (perhaps as nouns).

**Impact of Clustering Approaches.** As we already know from Section 4.8.3, clustering quality has big impact on SCD quality. Good clustering results usually lead to good SCD quality. For example, when the AFS technique was used in the clustering, the clustering quality was improved for *Reuter2k*. This also leads to good SCD quality in terms of F-score , which can be seen from Figure 5.9.

For hierarchical clustering, the SCD quality by different approaches exhibits similar pattern as the flat clustering. That is, the PagodaCD+ and the *CumulativeCD* approaches tend to produce higher quality SCDs in terms of F-score than other approaches. Figure 5.10 and 5.11 show the SCD quality for the first two levels of the hierarchical clustering for the *d5-1k* data set.

**Efficiency and Scalability.** Besides the clustering result, SCD is also a very important component for the SD. Although the construction of SCDs will require additional computational overhead, the benefit provided by the SCDs makes it worth it.

Relatively speaking, among those seven different approaches to construct SCDs, the PagodaCD+ approaches take more time than the others. Also, it usually takes

| ** Descriptive CD |
|---|
| Cluster 0 (232): engin(l)(227/589,0.0743) genet(229/662,0.0743) aspect(127/141,0.1906) seed(166/233,0.1271)<br>Cluster 1 (169): sludg(l)(135/253,0.0798) sewag(l)(103/176,0.084) irradi(l)(132/395,0.0873) fertil(l)(79/166,0.083)<br>Cluster 2 (269): standard(l)(171/300,0.1137) usda(S)(116/210,0.1131) regul(l)(75/120,0.1345) certif(l)(83/104,0.1787)<br>Cluster 3 (324): natur(l)(131/224,0.1253) food(l)(169/527,0.0749) irradi(l)(153/395,0.0873) label(l)(191/560,0.0757) |
| ** Discriminative CD |
| Cluster 0 (232): engin(l)(227/589,0.0743) aspect(127/141,0.1906) test(143/165,0.1783) seed(166/233,0.1271)<br>Cluster 1 (169): sludg(l)(135/253,0.0798) sewag(l)(103/176,0.084) engin(l)(94/589,0.0743) irradi(l)(132/395,0.0873)<br>Cluster 2 (269): engin(l)(94/589,0.0743) standard(l)(171/300,0.1137) genet(107/662,0.0743) food(l)(82/527,0.0749)<br>Cluster 3 (324): engin(l)(174/589,0.0743) aspect(7/141,0.1906) test(16/165,0.1783) seed(33/233,0.1271) |
| ** Frequency-Based CD** |
| Cluster 0 (232): genet(229/662,0.0743) engin(l)(227/589,0.0743) organ(l)(226/924,0.0715) product(l)(190/511,0.0767)<br>Cluster 1 (169): organ(l)(159/924,0.0715) sludg(l)(135/253,0.0798) irradi(l)(132/395,0.0873) genet(110/662,0.0743)<br>Cluster 2 (269): organ(l)(238/924,0.0715) standard(l)(171/300,0.1137) consum(S)(119/517,0.0754) usda(S)(116/210,0.1131)<br>Cluster 3 (324): organ(l)(301/924,0.0715) genet(216/662,0.0743) label(l)(191/560,0.0757) engin(l)(174/589,0.0743) |
| ** LN-Search CD (CDD) |
| Cluster 0 (232): meikl(4/4,0.2332) beltran(10/10,0.2332) english(9/9,0.2332) semant(3/3,0.2332)<br>Cluster 1 (169): enzym(12/16,0.1159) bioactiv(12/12,0.1698) engine(7/7,0.1698) amylas(12/12,0.1698)<br>Cluster 2 (269): kennedi(17/17,0.2704) reveal(18/18,0.2704) kathi(17/19,0.2178) tool(19/19,0.2704)<br>Cluster 3 (324): normal(13/13,0.3256) categori(36/42,0.2421) dna(16/17,0.2893) speci(16/18,0.2586) |
| ** Cumulative CD ** |
| Cluster 0 (232): aspect(127/141,0.1906) biologi(4/4,0.2332) english(9/9,0.2332) evan(10/10,0.2332)<br>Cluster 1 (169): alpha(6/6,0.1698) engine(7/7,0.1698) phase(8/8,0.1698) sludg(l)(135/253,0.0798)<br>Cluster 2 (269): arous(17/17,0.2704) fee(19/21,0.2243) standard(l)(171/300,0.1137) tool(19/19,0.2704)<br>Cluster 3 (324): dna(16/17,0.2893) normal(13/13,0.3256) organ(l)(301/924,0.0715) proper(17/19,0.2637) |
| ** Pagoda+ (CDD) |
| Cluster 0 (232): aspect(127/141,0.1906) engin(l)(227/589,0.0743) semant(3/3,0.2332) user(5/7,0.1279)<br>Cluster 1 (169): heavi(53/66,0.1154) format(6/7,0.1314) engine(7/7,0.1698) phase(8/8,0.1698)<br>Cluster 2 (269): tool(19/19,0.2704) confin(22/27,0.1844) written(15/17,0.2122) fee(19/21,0.2243)<br>Cluster 3 (324): natur(l)(131/224,0.1253) proper(17/19,0.2637) speci(16/18,0.2586) categori(36/42,0.2421) |
| ** Pagoda+ (CU) |
| Cluster 0 (232): moratorium(9/9,0.2332) english(9/9,0.2332) beltran(10/10,0.2332) evan(10/10,0.2332)<br>Cluster 1 (169): bioactiv(12/12,0.1698) amylas(12/12,0.1698) rennet(12/12,0.1698) cadmium(23/23,0.1698)<br>Cluster 2 (269): kennedi(17/17,0.2704) arous(17/17,0.2704) reveal(18/18,0.2704) tool(19/19,0.2704)<br>Cluster 3 (324): proper(17/19,0.2637) dna(16/17,0.2893) gene(33/35,0.2902) normal(13/13,0.3256) |

Figure 5.8: SCDs by Different Approaches for Data Set *d5-1k*

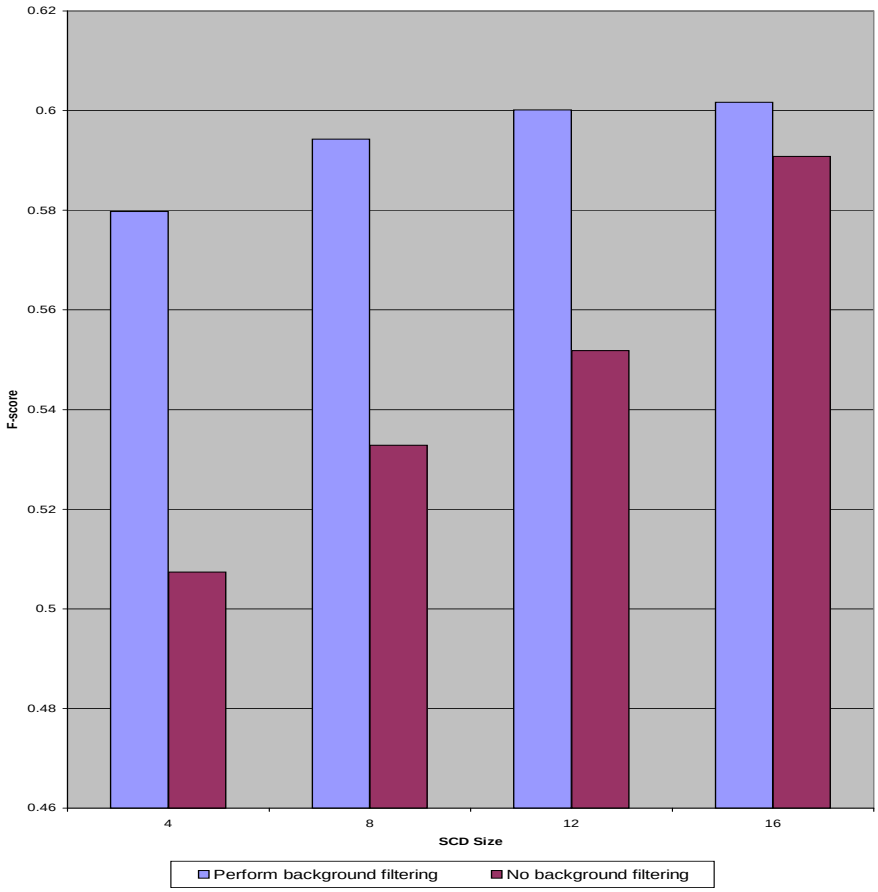



Figure 5.9: Impact of the ASF Technique on SCD Results for *Reuter2k* Data Set

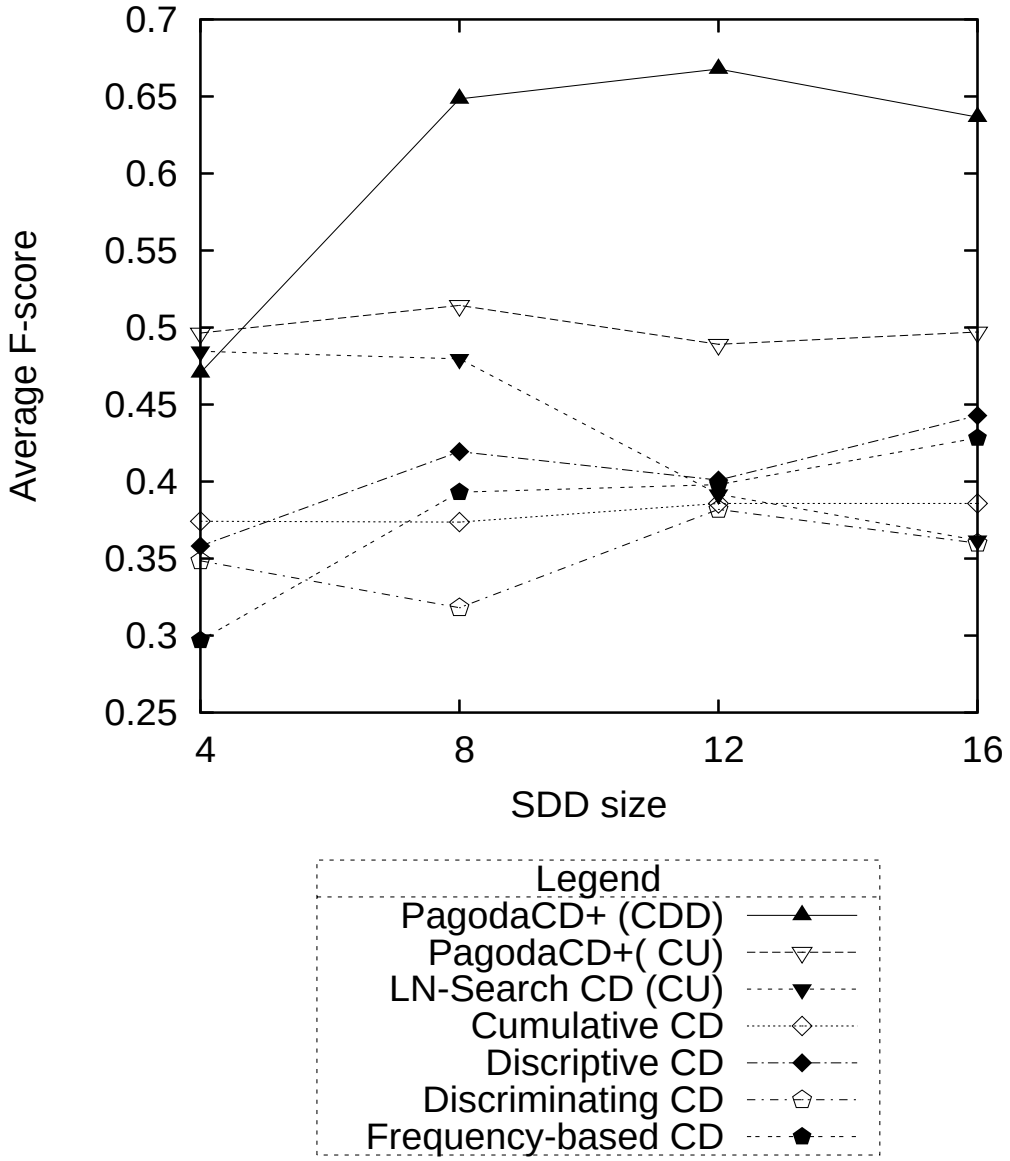



Figure 5.10: F-Score vs. SCD Size for Data Set *d5-1k* at Root-level of the Hierarchy

| | SCD Size | | | |
|---|---|---|---|---|
| | **4** | **8** | **12** | **16** |
| **Sub-cluster 1** | | | | |
| Descriptive CD | 0.58427 | 0.367635 | 0.333088 | 0.323746 |
| Discriminating CD | 0.545685 | 0.45485 | 0.396263 | 0.378923 |
| Frequency-based CD | 0.324526 | 0.324526 | 0.323746 | 0.323746 |
| LN-Search CD (CU) | 0.681438 | 0.577741 | 0.552061 | 0.630473 |
| Cumulative CD | 0.785049 | 0.830101 | 0.838202 | 0.845922 |
| PagodaCD+ (CDD) | 0.785049 | 0.81161 | 0.824195 | 0.824195 |
| PagodaCD+ (CU) | 0.66657 | 0.683725 | 0.689912 | 0.675396 |
| **Sub-cluster 2** | | | | |
| Descriptive CD | 0.512956 | 0.596836 | 0.385059 | 0.522078 |
| Discriminating CD | 0.591078 | 0.562721 | 0.414986 | 0.421928 |
| Frequency-based CD | 0.266616 | 0.240167 | 0.252835 | 0.252835 |
| LN-Search CD (CU) | 0.657368 | 0.544777 | 0.535882 | 0.509066 |
| Cumulative CD | 0.282422 | 0.282422 | 0.450217 | 0.590051 |
| PagodaCD+ (CDD) | 0.479649 | 0.637709 | 0.455303 | 0.58973 |
| PagodaCD+ (CU) | 0.614961 | 0.614961 | 0.59847 | 0.557852 |
| **Sub-cluster 3** | | | | |
| Descriptive CD | 0.816022 | 0.803904 | 0.752886 | 0.674946 |
| Discriminating CD | 0.69971 | 0.709316 | 0.600216 | 0.721014 |
| Frequency-based CD | 0.568769 | 0.595604 | 0.572066 | 0.597223 |
| LN-Search CD (CU) | 0.710987 | 0.688544 | 0.664575 | 0.698414 |
| Cumulative CD | 0.843947 | 0.875556 | 0.892229 | 0.892229 |
| PagodaCD+ (CDD) | 0.83171 | 0.902909 | 0.893772 | 0.912116 |
| PagodaCD+ (CU) | 0.661662 | 0.680961 | 0.72535 | 0.731187 |
| **Sub-cluster 4** | | | | |
| Descriptive CD | 0.399486 | 0.542815 | 0.336299 | 0.358518 |
| Discriminating CD | 0.346415 | 0.475317 | 0.368098 | 0.364295 |
| Frequency-based CD | 0.352454 | 0.364645 | 0.364638 | 0.366244 |
| LN-Search CD (CU) | 0.462321 | 0.535487 | 0.501975 | 0.593181 |
| Cumulative CD | 0.649414 | 0.686181 | 0.690749 | 0.701012 |
| PagodaCD+ (CDD) | 0.625437 | 0.656617 | 0.683665 | 0.687914 |
| PagodaCD+ (CU) | 0.50451 | 0.50916 | 0.478378 | 0.48728 |

Figure 5.11: F-Score vs. SCD Size for Data Set *d5-1k* at Second-level of the Hierarchy

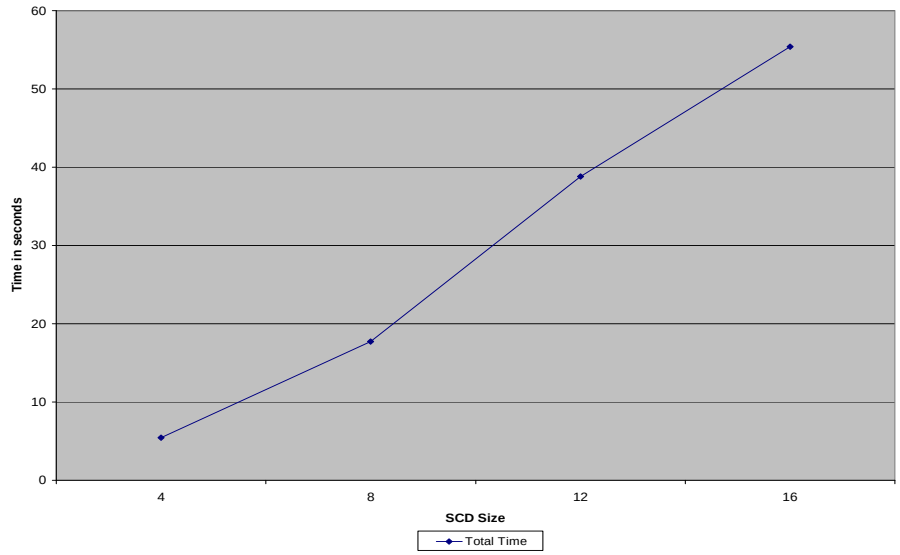



Figure 5.12: Total Time to Construct all 7 Different SCDs for Data Set *d5-2k* When SCD Size Changes

more time using the CDD measure than the CU measure. However, in terms of the actual execution time, they are all very efficient.

Figure 5.12 shows the total computation time in seconds to construct all seven different kinds of SCDs (i.e. Frequency-based SCD, Pagoda+(CDD) SCD, etc.) for the *d5-2k* data set. We can see that the execution time increases linearly when the SCD size increases. Figure 5.13 shows the computation time when number of documents in the data sets changes. Similarly, the time increases lineally when the size of the data sets increases. So we can see our approaches are efficient and scalable for large data sets[5].

[5]Even though we have not have time to apply our approaches to much larger data sets (i.e. millions of documents), we believe that our approaches can handle them if the machine has sufficient memory installed.

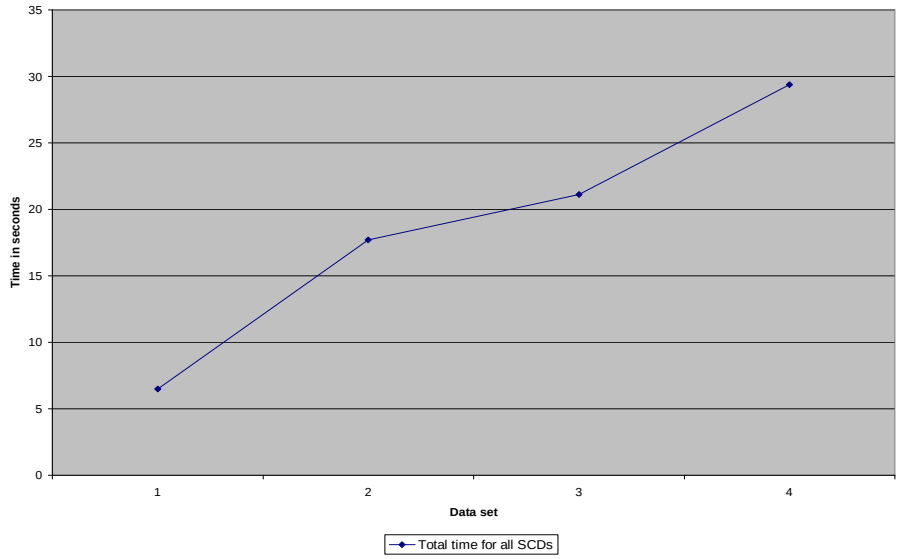



Figure 5.13: Total Time to Construct all 7 Different SCDs for Different Data Sets (1k, 2k, 3k, and 4k) When SCD Size is 8

## 5.6 Summary

In this chapter, we have introduced our novel approach to perform clustering and to construct SCDs simultaneously for large ERFRs. Our approach took consideration of three important aspects (namely opinions (O), issues (I), and stakeholders) of ERFR throughout the process. We believe that the clustering result, along with the informative SCDs, can be an excellent navigation aid for the rule-writers and analysts to digest large ERFRs effectively and efficiently.

For the clustering, we have proposed to use our *active feature selection* (AFS) and *adaptive similarity measure* (ASM) approaches. In the AFS approach, we utilized the proposed rule and general-topic document collection as *background knowledge* to help perform feature selection. In addition, we also add O, I and S terms into the feature set. In the ASM approach, we proposed the OIS-TFxIDF model to adjust the feature weights to meet different needs of users. In this model,

users can put more emphasis on O, I and S depending on the situation.

For the SCD construction, we have provided additional quantitative information to enhance the CD terms proposed in Chapter 4. In addition to the CDD measure, we also tried to use the CU measure as another surrogate measure for searching quality SCDs. We proposed a layer-based replacement search algorithm called PagodaCD+ , which is an enhanced version of the PagodaCD algorithm introduced in Section 4.6.2. The PagodaCD+ algorithm can utilize any improvement measure and can also obtain additional statistical information for the SCDs.

We have conducted experiments on several publicly available e-rulemaking data sets. Experimental results demonstrated that our methods can produce high quality clustering and SCDs for given ERFRs.

While our work showed promising results in digesting large ERFRs, we believe that the quality of this work can be further improved if we can do the following: (1) utilizing some domain-specific ontology as background knowledge in the clustering process, (2) exploiting other important aspects for managing large ERFRs, (3) studying ways to make the SCDs more conceptually coherent across the clustering, (4) involving more human evaluation efforts to further validate the useability of the clusterings and the understandability of SCDs, (5) replacing the root terms to non-root form, and (6) generalizing the idea to provide high-level "map" for other large repositories.

In the next chapter, we will present our approach to selecting representative arguments (RAs) for the ERFR clustering. The RA is also an important part of the summaritive digest (SD) for large ERFRs.

# Chapter 6

# Selecting Representative Arguments for ERFR Clustering

Previously, we introduced our *OIS-based* approach to perform clustering and also construct SCDs for given ERFR at the same time. Experiments demonstrated that the *OIS-based* approach can produce quality clustering and informative SCDs, which are good navigation aid for the rule-writers to digest the underlying feedbacks effectively.

In this chapter, we will present our approach to selecting some *representative arguments* (RAs) for each cluster of a given ERFR clustering. As mentioned in earlier chapters, the RAs are an important part of the summaritive digest (SD) for managing large ERFRs. The RAs for each cluster can give the rule-writers some representative "tastes" about the underlying feedbacks.

## 6.1 Introduction

When facing large amount of documents, such as ERFR, it is a great challenge to manage and digest them effectively and efficiently. There have been many

studies on document clustering to address the problem, which were discussed in previous chapters. There are also increasing number of works on constructing CDs, including our work introduced in Chapters 4 and others such as [67]. Document clustering and SCD construction can help the users on organizing and labeling the large document repositories.

In addition to the helps offered by clustering and SCDs to the users, however, it is also very desirable to select some representative arguments (RAs) for each cluster of a given clustering so that users can have additional way to sense the main theme of the underlying collections. It is especially the case for managing large ERFRs. We believe that carefully selected RAs can serve this purpose. Therefore, rule-writers may taste most of the flavors of the underlying feedbacks just from the RAs. In addition, since RAs are fluent sentences directly extracted from the original feedbacks, they are easier for the human users to comprehend. In this sense, RAs will be a better aid than SCDs for the rule-writers to digest the giving ERFR.

There have been efforts to extract sentences from a single document and multiple documents to produce a summary. We briefly surveyed some of the works in Section 2.3.1.1. However, the goal of those approaches is trying to summarize the document(s) in the sense to "compress" the document(s), not to select typical *arguments*. In addition, those approaches consider all the sentences in documents during summarization, and do not address the special needs for managing ERFR.

In this research, we study ways to select good RAs for ERFR clustering. As mentioned earlier, RA is an important part of the overall summaritive digest (SD). We believe that the RAs should not only reflect the arguments contained in large portions of the commentators, but also some typical views hold by small portions of them. In other words, we want the RAs to represent diversified opinions from the feedbacks so that different "voice" can be heard.

We believe that sentences in which major stakeholders have expressed some opinions on some of the important issues might be the most important sentences in feedbacks. Therefore, one of the important considerations of this work is that we only consider such sentences, which we refer to as *arguments*, as candidates for representative arguments.

In this work, we propose to use our *RAPDC* approach for selecting RAs. Roughly speaking, the *RAPDC* approach first identifies and extracts all the arguments from each feedback based on those identified I, O and S terms. Then, the orientation of those candidate arguments is determined. After the clustering was produced, a desired number of RAs will be selected for each cluster based on three important factors, *popularity*, *diversity* and *CC-coherence* .

In the following sections, we will first give an overview of some related works. Then we will introduce our approaches for (i) identifying and extracting augments, (ii) determining the orientation of each argument, and (iii) selecting representative arguments for each cluster. We will show some experimental results and our quality evaluation strategy before concluding this chapter.

## 6.2 Related Works

In the document summarization research community, there have been a lot of works to extract sentences from documents to form a summary [28, 55, 39, 101, 85]. Some of the well-known summarizers using this kind of approach are MEAD, WebSumm, SUMMARIST and LEAD. See Section 2.3 for a brief overview of those approaches. Our *RAPDC* approach is different from those in two ways. First, our approach only considers certain sentences, which we call *arguments*. We do not consider all the sentences like the other approaches do. Second, unlike those summarizers, our approach tries to select some representative arguments that users are interested

in, and it does not intend to summarize the whole content of the document(s).

Recently, there have been a lot of attention on identifying opinions from product reviews and weblogs (or blogs) [90, 86, 22, 56]. For example, Opinmind[1] is a commercial weblog search engine which can categorize the search results into *positive* and *negative* opinions. References [90, 22, 56] try to classify user opinions (e.g. thumbs-up or thumbs-downs) for certain product or product features. In [86], authors proposed a probabilistic model to identify topics and sentiments simultaneously for a given weblog collection. In this work, we only interested in those opinions expressed by some major stakeholders on some of the important issues. For those qualified sentences, we also want to identify their sentiment. In addition, we also select some representative sentences for each cluster.

The needs for selecting representatives are everywhere. We now use analogies from other situations to discuss some desired properties on representatives. When selecting representatives, we usually want them to represent the underlying population well, and also be diversified. Interestingly, there are often trade-offs between those two criteria. In addition, it is often domain dependent on how to measure the representativeness and diversity, and how much emphasis needs to be put on each factor. For example, when determining which stocks to invest, people usually want to diversify their investment by looking for those stocks that have good potentials in different sectors. In this case, the diversity may be more important than representativeness. However, assume that only the stocks in the energy sector perform well, people may invest more to this sector. Therefore, trade-off occurs when sacrificing some diversity to gain more representativeness (currently good performers). These two criteria also apply to our work to select RAs.

There have been different ways to measure diversity depending on the domain. For example, we define diversity in terms of document overlap in Section 4.5.

[1] http://www.opinmind.com

Reference [66] uses Q-statistics to measure the diversity for classifier ensembles. Biologists often use entropy-like model to choose diversified set of species [98, 62]. In this work, we consider diversity based on the three important aspects (i.e. O, I and S) of ERFR clustering.

## 6.3 Extracting Arguments From ERFR

Recall that our goal is to select some representative sentences for each cluster of a given ERFR clustering. Therefore, the first major step of our approach is to extract all candidate sentences from the feedbacks. Below, we will discuss what kind of sentences are considered as candidates.

### 6.3.1 Definition of ERFR Argument

In general, each feedback contains some opinion expressions that are related to the proposed rule. However, not all the opinions are expressed by those major stakeholders. In addition, the major stakeholders may have expressed opinions on some other issues that we are not interested in. In this work, we only focus on those opinion expressions that major stakeholders have expressed on some of the important issues. If a sentence contains such opinion expression, we call it *argument*. The concepts of *stakeholders*, *issues* and *opinions* were discussed in Chapter 3, and the O/I/S-terms should be already extracted before the RA selection.

**Definition 6.3.1** An *argument* is a sentence in which some major stakeholders have expressed opinions on some of the most important issues. ■

**Example 6.3.1** *Using the DoA-NOP data set as illustrated in Section 3.3.3 as an example, assume that "consumer" and "usda" are two major stakeholders, "rule" and "sewage sludge" are two important issues. Lets consider the two sentences*

*given below in feedback A.1.2 (this feedback is listed in Appendix A in its entirety). We can see that the following sentence will be considered as an argument.*

> *I am outraged as a consumer that irradiated foods, genetically engineered foods, and foods grown on lands fertilized with sewage sludge are included in the USDA's proposed rules for organic foods...*

*However, the following sentence will not be considered as an argument.*

> *I enjoy the fruits of a 38 acre burgeoning farm in the Blue Ridge mountains of Virginia...* ■

Note that stakeholders could express different opinions on one or more issues in one argument.

### 6.3.2 Identifying Arguments

Having defined what kind of sentences we are interested in, e.g. the *arguments*, we now discuss how to identify and gather them.

Roughly speaking, we use a technique of shallow syntax parsing to identify the arguments. The parsing process will reference some of the already identified O, I and S terms of the ERFR. The overall process for identifying arguments is illustrated in Figure 6.1.

First, we process the feedbacks to produce the Part-Of-Speech(POS) tags. Then, we identify opinions, stakeholders and issues. Finally, we extract all the arguments based on the tagged sentences and those identified O, I and S terms. In Chapter 3, we briefly discussed those steps; recall that the identified arguments were extracted and saved to a file as described in procedure 3.4.1.

Note that when identifying arguments, we treat all personal pronouns (i.e. "I", "we", "you", etc.) as if they are major stakeholders. By "expanding" the

stakeholders we can pick up more arguments. Indeed, we view all the feedbacks as submitted by stakeholders, either a person or an entity, which probably are also major stakeholders.

Also note that we consider those sentences that contain cue phrases as arguments. We utilize several cue phrases, such as *? oppose, ? support, ? suggest*, etc., where the "?" represents pronouns.

### 6.3.3 Determining Argument Orientation

For each identified argument, we also want to determine its orientation. The orientation will be used as a factor in the representative selection process.

Determining the orientation of an argument is a little bit more involved than determining that of an individual word, since there could be multiple, sometimes opposing, opinion words in an argument. In this work, we assume that there are two sentiment polarities for each argument if the sentiment is known, i.e. the *positive* and the *negative* sentiment.

Our approach is to simply use the voting mechanism based on the orientation

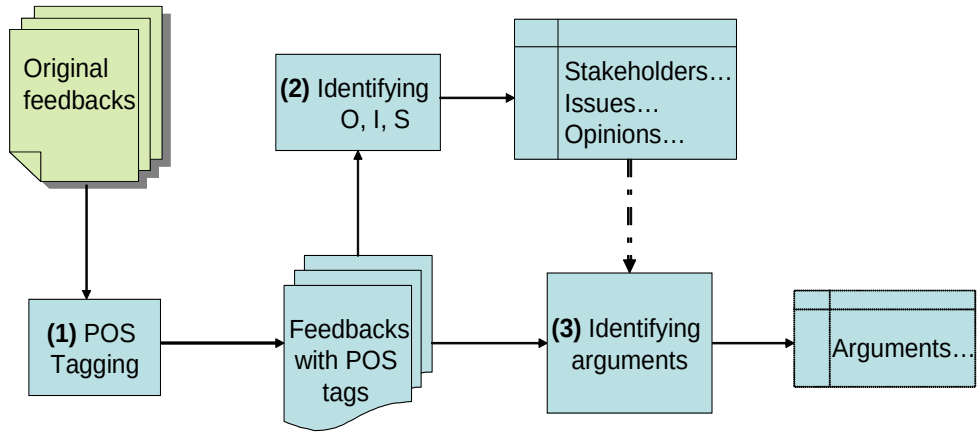



Figure 6.1: Identifying Arguments

of all the opinion words appearing in an argument. That is, if *positive/negative* opinion prevails, the argument is regarded as a *positive/negative* one. Otherwise, it will be treated as *unknown*.

Note that if there is a negation word such as "no", "not" appearing around the opinion word, we treat the orientation to be the opposite of the orientation of the opinion word itself. In this work, we set the maximum word distance between the negation and the opinion words to be 2. The negation word can be before or after the opinion word. For example, "we do not support ..." will be considered as *negative*, since the word "support" is considered as *positive*. We consider the distance between "not" and "support" to be 0. In another example, the argument "In no way the genetically engineered food is safe" will be considered as positive which is a mistake. This is because distance between "no" and "'safe" is 6, which exceeds the threshold 2.

We also consider the situation if a sentence that contains a sentimental change sub-clause, such as *but, however*, etc. In such case, we first compare the aggregated opinions between the main clause and the sub-clause. Then, we treat the overall orientation be the opposite to the winning side[2]. For example, "this proposed rule sounds good, but I don't buy it." will be considered as *negative*. In this example, the sentiment of the main clause is *positive* because of the positive opinion word "good". However, because of the sentimental change sub-clause "but", the overall sentiment is considered as *negative*.

## 6.4 Selecting Representative Arguments (RAs)

We now discuss our approach to selecting representative arguments (RAs) for each cluster of a given ERFR clustering.

[2]If there is no opinion words in the main/sub-clause, the winner will the sub/main-clause.

When arguments were identified and extracted, each argument itself along with its orientation are saved in a file. After clustering was performed, the associated clusters for each argument are also collected. Now, our goal is to select some small number $k$ of arguments for each cluster as representatives. We hope that those RAs can pick up important, but also different, "voice" from the underlying feedbacks. In other words, we want RAs not only to contain some of the most prevailing opinions, but also some not-so-popular, yet important, opinions.

As an implementation note, after RAs were selected for each cluster, users have the choice whether or not to show the RAs, or how many they want to see at a time (default to 5). In addition, only portion of the RAs will be displayed initially, and users can click the underline hyperlink to see the complete argument.

Below, we will first discuss three important factors that will be considered when selecting RAs. Then we will introduce our approach, namely the *RAPDC* approach, for selecting RAs.

### 6.4.1 Important Factors for Choosing RAs

We consider *popularity*, *diversity* and *cross-cluster coherence* (short handed as *CC-coherence*) as three important factors for selecting RAs.

Intuitively, *popularity* is used to encourage the selection of those popular arguments that are stated by a large number of the commentators; *diversity* is used to encourage the selection of those unique arguments that are not so popular, yet are important, among the commentators; and *CC-coherence* is used to ensure that RAs selected for each cluster are indeed good representatives for that cluster by looking at a broader picture - the whole clustering. We use these three factors to help us to capture the quality needed for good RAs.

We now discuss each of the factors in turn. Let $\mathcal{A}_i = \{A_{i1}, A_{i2}, ..., A_{ir}\}$ be a collection of $r$ arguments for cluster $C_i$. For simplicity, we omit the cluster

index $i$ for each argument when there is no ambiguity on the target cluster. i.e. we use $A_j$, instead of $A_{ij}$, for the $j_{th}$ argument of cluster $i$. For each argument $A_j$ ($1 \leq j \leq r$), we have the cluster information ($i$), the argument orientation ($\mathcal{O}_j$, which could be *Positive*, *Negative* or *Unknown*), and a bag of words for the argument ($\mathcal{W}_j$). Therefore, the associated information for argument $A_j$ can be expressed as $< i; \mathcal{O}_j; \mathcal{W}_j >$, where $\mathcal{O}_j \in \{P, N, U\}$, and $\mathcal{W}_j =< w_{j1}, w_{j2}, ..., w_{jm} >$ consisting of $m$ unique words.

#### 6.4.1.1 Popularity

*Popularity* of an argument in a given cluster is used to measure how popular the argument is in the cluster. A more popular argument will be a better choice for an RA.

Given $r$ arguments for a cluster, the challenge is how to find out which argument(s) is (are) more popular than others? It is not trivial, since people may use slightly different languages to express similar idea.

Since the orientation of each argument has been identified at this point, we will consider the *popularity* for each orientation category, i.e. $\mathcal{P}_P$ for *positive*, $\mathcal{P}_N$ for *negative*, and $\mathcal{P}_U$ for *unknown*. By considering them separately, we can answer more detailed questions, such as "what are the typical arguments in favor of/ against the rule?". Ideally, the opinion orientation within one cluster should be the same, but there are exceptions.

For arguments that belong to the same orientation category, we first perform clustering on them based on the similarity of their word set $\mathcal{W}$, so that similar arguments will be grouped together. Then, we can find the popular arguments from the "center" of those clusters (the center of a cluster is the argument whose corresponding vector representation is most similar to the centroid vector of the cluster with respect to the cosine similarly measure); the number of arguments

selected from each cluster will be proportional to the cluster size. Note that *argument clustering* is different from *feedback clustering*. The purpose of argument clustering, a mini-clustering, is try to group similar arguments together so that we can find out about the popular arguments. Also note that by selecting arguments from different argument clusters, we could pick up some arguments that have different perspectives, but they are still very similar overall.

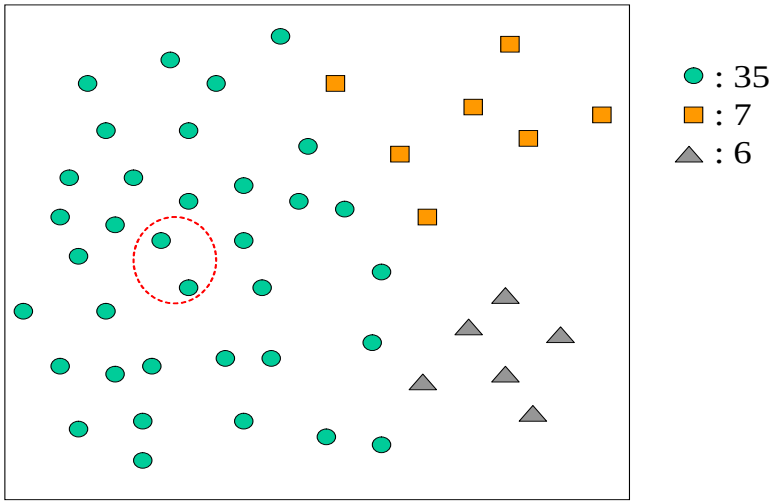



Figure 6.2: Illustration About Three Clusters of Arguments

Figure 6.2 illustrates how 48 arguments are clustered into three clusters, with cluster sizes of 35, 7 and 6, respectively. Assuming only 2 popular arguments will be selected among them, those two arguments will be all from the largest cluster, the "circle" cluster, because of the size differences of the clusters. The choice could be the two arguments as indicated in the dash-circle.

We now briefly discus how the clustering will be performed. The clustering is only for the candidate arguments, not all the sentences. We use the efficient bisecting k-means algorithm as discused in Algorithm 5.3.1. In addition, the number of clusters will be usually small, such as 2 or 3. Therefore, the extra cost is

not significant. We use the cosine similarity measure for the clustering. That is, for argument $A_i$ and $A_j$, the cosine measure is defined by the cosine of the angle between the two word vectors $\mathcal{W}_i$ and $\mathcal{W}_j$ associated with the two words:

$$sim_{cos}(A_i, A_j|\mathcal{O}) = \frac{\mathcal{W}_i \cdot \mathcal{W}_j}{\|\mathcal{W}_i\| * \|\mathcal{W}_j\|}.$$

Note that the similarity measure was based on the given argument orientation $\mathcal{O}_j$, where $\mathcal{O}_j \in \{P, N, U\}$, because argument clustering was performed for each orientation category.

#### 6.4.1.2 Diversity

While we emphasize the *popularity* factor when selecting RAs, we are also interested in presenting some diversified arguments to the user so that different "voice" can be heard.

*Diversity* is used to measure how different those selected arguments are. An argument that is more different from the existing choices will be a better addition to the RAs.

Let's consider the same example as illustrated in Figure 6.2. Based on the *diversity* criterion, we should choose arguments one from each of the clusters, instead of all from the same "circle" cluster.

The importance of *diversity* can be seen from many applications as discussed in Section 6.2. In general, when the objects (RAs here) are similar to each other, it is hard to synthesize the entire cluster from them; in contrast, when they are more widely and evenly distributed, they can be combined to offer better picture of the whole cluster.

Now, the hard question is what kind of arguments are considered as diversified? The answers may vary depending on the criteria used. In this work, we consider the *diversity* factor along the following two directions:

- Different orientation. If two arguments have different orientation, then those two arguments will be considered as diversified. In this case, we try to choose certain number of arguments from each orientation category that is proportional to the number of candidates in that category. For example, if a given cluster has $p$ *positive* arguments, $n$ *negative*, and $u$ of *unknown*, then the percentages to choose from each of the categories are $p/(p+n+u)$, $n/(p+n+u)$ and $u/(p+n+u)$, respectively.

- Same orientation. Among arguments with identical orientation, the diversity will be determined by the dissimilarity of their content. In this case, we select arguments from different argument clusters. Recall that argument clustering was performed based on their word vectors as discussed in Section 6.4.1.1. The intuition for this is that "far way" arguments are more diversified than "more closer" arguments. For example, suppose there are already some existing RAs, denoted as $AE$. For any other argument $A_i$, the diversity (denoted as $\mathcal{D}$) is measured using $\mathcal{D}|\mathcal{O} = (1 - \sum_{A_j \in AE} sim_{cos}(A_i, A_j|\mathcal{O}))$. The larger the $\mathcal{D}|\mathcal{O}$ value, the more diversified they are. ■

#### 6.4.1.3 CC-Coherence

So far we have considered the *popularity* and *diversity* factors for each individual cluster separately. A new issue arises: there might be inconsistency when looking at the RAs across the entire clustering.

We introduce the notion of *CC-coherence*, in which CC stands for cross-cluster, to address the inconsistent situation for the selected RAs. We can adjust[3] the RAs by considering the whole clustering after they have been selected by only considering situations within individual clusters. In this work, we consider the following inconsistent situations:

[3]The adjustment step can improve the overall RA quality, but it is optional.

- When RAs from different clusters are very similar. In this case, those similar RAs may cause wasted bandwidth. Therefore, there is a need to reduce the "overlap" conveyed by those RAs. In this work, we try to adjust the RAs of those clusters with more arguments candidates, by replacing those overlapping RAs with other popular or diversified arguments.

- When RAs do not seem to be good representatives for the given cluster. There are some cases that an argument may be appealing, but it may not fit well in the current cluster if we consider them together within the entire clustering. For example, when the main theme of a cluster is "against" some issue, an RA that is "in favor" will be considered as unfit, even though the argument is strong. In this case, we want to move this argument to the cluster with the theme on "in favor" of this issue.

- When the clustering is very skewed. Some clusters may have many more feedbacks than other clusters, or certain clusters may have much fewer feedbacks compared to others. In this case, we can adjust the number of RAs for those clusters, especially for those large clusters. ■

To help determine whether an argument is good for a cluster, we can reference the score of $P(\mathcal{W}_i|C_t)$, where $C_t$ denotes the given cluster and $\mathcal{W}_j =< w_{j1}, w_{j2}, ..., w_{jm} >$ is the word vector for the argument $\mathcal{A}_i$ of cluster $C_t$.

Based on Formula 5.3, and also assuming that words in arguments are independent, i.e. $P(\mathcal{W}_i) = \sum_{d=1}^{m} P(w_d)$, we will have Equation (6.1):

$$P(\mathcal{W}_i|C_t) = \frac{1 + \sum_{j=1}^{|\mathcal{A}|} \sum_{d=1}^{m} tf_{(d,j)} P(C_t|A_j)}{|T| + \sum_{s=1}^{|T|} \sum_{j=1}^{|\mathcal{A}|} tf_{(s,j)} P(C_t|A_j)} \tag{6.1}$$

In this equation, $\mathcal{A}$ denotes all the arguments for the clustering, $|T|$ is total number of terms that appeared in the arguments, $tf_{(i,j)}$ is the frequency of term $w_i$ in

argument $A_j$, and $P(C_t|A_j) \in \{0, 1\}$ is the probability of cluster $C_t$ given $A_j$ (which depends on the cluster membership of the argument $A_j$).

In this work, we check those inconsistent situations by looking through all RAs across the clusters and also referencing the $P(\mathcal{W}|C)$ values. Some adjustment will be made manually if found necessary. Note that those "check-adjust" steps could be repeated so that the quality of RAs can be improved across the entire clustering.

### 6.4.2 RAPDC: The Proposed Approach

We now turn to our approach for selecting the RAs. Our proposed approach will be based on the three factors of *popularity*, *diversity* and *CC-coherence*, as discussed previously. We call our approach *RAPDC*, which stands for RAs selection based on pularity, diversity and CC-coherence.

When dealing with multiple factors, there are trade-offs. In this case, there is trade-off between the *popularity* and *diversity*. In the RAPDC approach, we consider *popularity* and *diversity* simultaneously by giving them appropriate shares, i.e. we let certain number of RAs be more popular, while others are more diversified. The *CC-coherence* factor will be considered after the RAs have been initially selected based on other two in-cluster factors, *popularity* and *diversity*.

For a given clustering $\mathcal{C} = \{C_1, ..., C_l\}$, suppose there are totally $m_i$ arguments extracted from a given cluster $C_i$ and we want to select $k_i$ RAs for the cluster. In general, the number of RAs for each cluster will be same, i.e. $k_1 = k_2 = ... = k_l$. Our *RAPDC* approach works as follows:

---

**Procedure 6.4.1** RAPDC Approach for Selecting RAs

1. *Identify and extract all candidate arguments as discussed in section 6.3.2;*

2. *Determine the sentiment of each argument as discussed in section 6.3.3;*

3. *Associate cluster indexes to each argument after the clustering is formed;*

4. *For each cluster $C_i$ ($1 \leq i \leq l$), select RAs by considering the* popularity *and the* diversity *factors. For the goal of total $k_i$ RAs for each cluster, we select $\lceil \lambda k_i \rceil$ of RAs to satisfy the* popularity*, while $\lfloor (1-\lambda)k_i \rfloor$ of them will be used to address the* diversity *issue. The $\lambda$ is a user specified parameter to address the importance of each factor. The larger the $\lambda$, the more emphasis is given to the* popularity *factor. We use $\lambda = 0.6$ as default.*

   (a) *Popularity. The set of all $m_i$ candidate arguments will be clustered as discussed in section 6.4.1.1. The number of clusters is usually small depending on the total number of arguments, e.g. 2 or 3. Then $\lceil \lambda k_i \rceil$ of RAs will be selected according to the argument clusters.*

   (b) *Diversity. $\lfloor (1-\lambda)k_i \rfloor$ of RAs will be selected to address the* diversity *as discussed in section 6.4.1.2.*

5. *Check the cc-coherence of the RAs in the view of entire clustering, and make necessary adjustment as discussed in section 6.4.1.3. This step can be repeated if necessary.* ■

---

By considering those three factors collaboratively, we can obtain good quality RAs for the clustering of ERFR.

## 6.5 Experimental Evaluation

We now discuss how to evaluate the quality of the RAs. It is always a challenging task to evaluate any type of document related results, especially for large-scale evaluation. It will be desirable if we can find some objective ways/measures to automate the evaluation. However, in most cases human judgment is unavoidable.

Below, we first discuss the evaluation methodology that we use, then we present some experiment results and examples.

### 6.5.1 Evaluation Methodology

Recall that the purpose of the RAs is to present some typical arguments of each cluster to the rule-writers, so that they can have some "taste" about the broader arguments in the underlying feedbacks. Therefore, the evaluation will need to be focused on validating whether such goal has been achieved.

In addition, the quality of RAs should be considered in an extended context, i.e. should be looked at in connection with the clustering quality and the SCD quality, since RAs is only a part of the summaritive digest (SD) for large ERFR.

Having those objectives in mind, we employ human efforts to evaluate the RAs, even though human judgment is time-consuming, expensive and perhaps inconsistent. Clearly, large-scale evaluation is very hard, if not impractical.

In this work, two people, acted as judges, went though same amount of the RAs that selected for certain clusters. For simplicity, we let the judges use binary scale to categorize each argument: *good* (the argument clearly reflect the main theme of the cluster, or it represents a diversified point of view), and *bad* (the argument is not a good *argument* based on our definition, or it is not a good representative for that cluster). The quality of RAs, called *goodness*, will be measured by the average value on the percentage of *good* arguments rated by the two judges.

To measure the agreement between two judges, we use the Cohen's *Kappa* measure[4], which is designed for categorical judgments and can take into account the agreement that occurs by chance. *Kappa*, denoted as $\kappa$, is defined as

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)},$$

[4]http://en.wikipedia.org/wiki/Cohenś_kappa

where $P(A)$ is the proportion of the time the two judges agreed, and $P(E)$ is the proportion of the time they would be expected to agree by chance. $\kappa$ has value between 0 and 1, i.e. $\kappa \in \{0, 1\}$. Larger $\kappa$ means better agreement, which implies higher confidence on the rating of RA quality. Specially, when $\kappa = 0$, it means the agreement is by chance, and when $\kappa = 1$, there is total agreement. Also, it is commonly regarded that when $\kappa \geq 0.8$ it means good agreement, and $0.67 \leq \kappa < 0.8$ means "tentative conclusions" [30].

$P(E)$ is usually estimated by $P(E) = P(good)^2 + P(bad)^2$. In practical, the value $P(E) = 0.5$ is usually used for efficiency purposes.

### 6.5.2 Experiment Results

In this experiment, we mainly use the data sets *d5-50* and *d5-1k*, which were discussed in Section 5.1. Also, we try to select five RAs for each cluster by default, unless specified otherwise.

As discussed in Section 6.4.2, RAs are selected based on the Procedure 6.4.1. The procedure works for both flat and hierarchical clusterings. Note that the candidate arguments are identified by Procedure 3.4.1. Some examples of the candidate arguments for data set *d5-1k* are given in Appendix B.2.

**Flat Clustering.** Figure 6.4 lists the RAs selected for the data set *d5-1k*, where the feedbacks were grouped into four clusters when the OIS factors were emphasized together. Figure 6.3 illustrates the meaning of the format for the RAs. In this example, we selected five RAs for each cluster in the clustering, two of which have "positive" orientation, two "negative and one "unknown". From the document indexes, we can see that the arguments are very diversified based on their origin, i.e. scattered across the collection.

**Hierarchical Clustering.** Figure 6.5 shows the first two levels of the hierarchical RAs for data set *d5-50*. The actual clustering structure can be seen from Figure

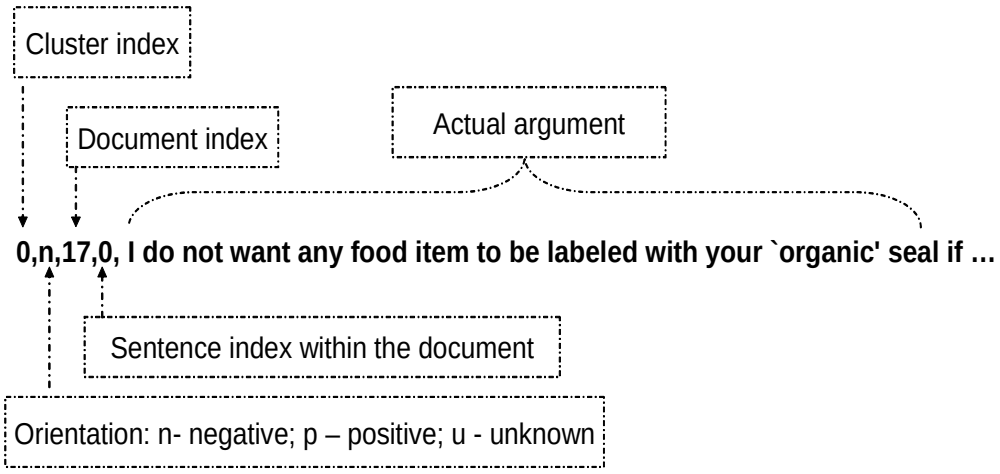



Figure 6.3: The Format for the Selected RAs

5.5. We can see that most of the RAs for cluster 0 have negative orientation; RAs for cluster 1 are mixed, but majority of them are positive.

Note that while the RAs maintain the overall theme of the underlying clustering (i.e. negative vs. positive), the arguments are also selected from diversified indexes. However, there are two issues that are worth mentioning here. First, the orientation of the first RA for cluster 1 is negative. Based on the *CC-coherence* factor, we could replace this RA with other non-positive arguments. Second, the RAs for cluster 0.0 are all selected from two different feedbacks, and it seems they are not diversified in terms of feedback origin. However, if we look at the actual clustering, the results actually make sense. This is because only five feedbacks are in this cluster, three of which are duplicated and the other two of which are very similar to each other.

**Quality of The RAs.** Recall that the purpose of the RAs is to provide some typical, yet diversified, arguments for each cluster so that users can get some "taste" about underlying feedbacks. Therefore, the quality assessment should be based on whether or not such goal can be met.

| |
|---|
| 0,n,326,3, Because of the present, ill-advised decision not to require such labeling, those consumers who choose not to eat these potentially dangerous products have turned to organic foods because they are free of genetic engineering.<br>0,n,541,0, Please do not allow genetically-engineered or irradiated foods to be labeled as `` organic.<br>0,p,290,8, You will be honored in history for your foresight in creating a safe haven.<br>0,p,466,1, All genetically engineered foods should be labeled as such so that I may have the right to an informed choice as to whether I choose to consume these foods.<br>0,u,34,0, I feel it is imperative that Genetically engineered foods be disallowed in all organic compounds. |
| 1,n,156,3, I 'm also very concerned about the fact that irradiating foods kills the vital life forces within those foods, and do not want to see this practice allowed under the guidelines.<br>1,n,884,3, Irradiation is not a natural process and should not be allowed under these standards.<br>1,p,21,1, I support keeping the use of sewage sludge out of the final regulations.<br>1,p,943,12, When I feed my family organic food ( which is about 80 % ) I feel good knowing I 'm doing my part to keep them and the planet healthy.<br>1,u,382,1, I would want to know about any pesticides or commercial fertilizer used in growing food as well as the things I have noted above. |
| 2,n,107,28, We must reject this meretricious abuse of language and insist on truthful labeling.<br>2,n,45,8, The proposed organic food standards will bring about very bad results.<br>2,p,774,0, Hi, I think its great that the federal government is getting on the organic foods bandwagon.<br>2,p,818,1, It is bad enough that consumers are forced to pay premium prices to ensure our food is safe from pesticides, hormones and other dangers; the proposed standards would now hoodwink consumers into paying premium prices for products they believe are safe and organic which may be neither.<br>2,u,524,3, The entire program should be funded through an alternative source rather than on the backs of the small organic producer. |
| 3,n,786,1, I do not like the idea of including irradiation techniques or sludge ( which might contain toxics, chemicals ) in the definition of organic.<br>3,n,856,0, I think the USDA is making a serious mistake in even considering the use of bioengineering, irradiation, sewage sludge and antibiotics in the organics industry.<br>3,p,793,3, Please do not produce a standard that commercial organizations with no philosophical or health interest in organics and food safety can use to masquerade as, and compete with, organizations that do fully support and participate in the organic foods movement.<br>3,p,800,0, Comprimise of pure organic standards is not acceptable.<br>3,u,555,27, Products proven to be free of GEOs may also be labeled as GEO free. |

Figure 6.4: RAs for Data Set *d5-1k* When the Number of Clusters is 4

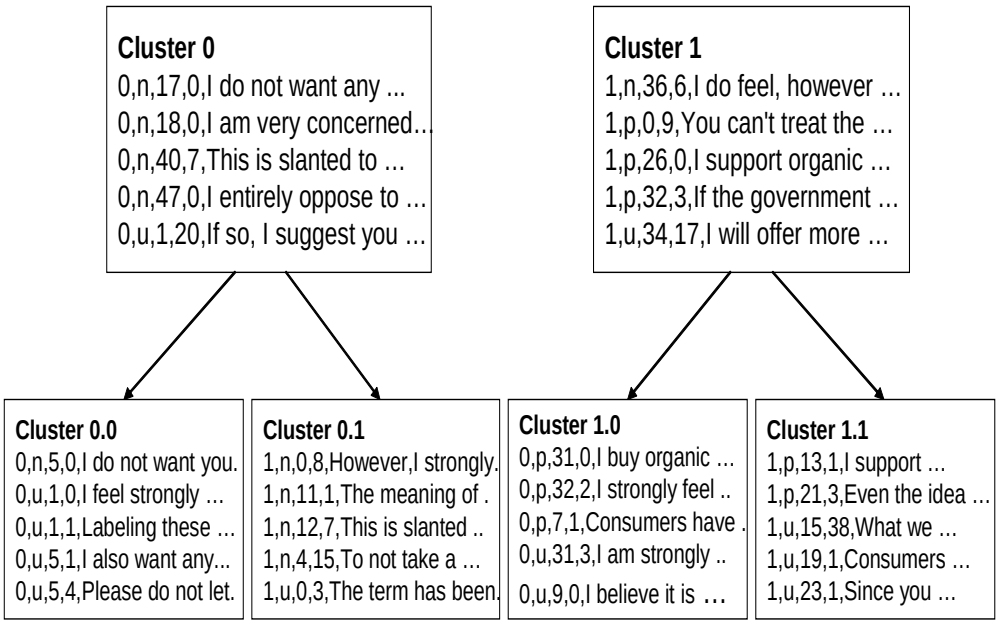



Figure 6.5: Hierarchical RAs for Data Set *d5-50*

In this work, we did an evaluation using the smaller data set *d5-50*. First, two person[5] read through all 50 feedbacks, and they were also given the clustering structure and the RAs, which are illustrated in Figure 5.5 and 6.5, respectively. Then, for each selected RA (from a total of 30), it was categorized into one of the two cases: *good* or *bad* as mentioned in Section 6.5.1.

The evaluation results are shown in Table 6.1. From the results, we have $P(A) = (26 + 2)/30 = 0.93$. Since $P(good) = (26 + 26 + 1 + 1)/60 = 0.9$ and $P(bad) = (2 + 2 + 1 + 1)/60 = 0.1$, so the estimated $P(E) = 0.9^2 + 0.1^2 = 0.82$. Therefore, $\kappa = (0.93 - 0.82)/(1 - 0.82) = 0.61$, which indicates two judges don't have very good agreement. The *goodness* of the RAs can be measured as $\frac{((26+1)/30+(26+1)/30)}{2} = 0.9$.

From the above $\kappa$ value, we can see that human evaluation is very subjective

[5]My wife Chunxia helped a lot in this case.

| # of RAs | Judge 1 | Judge 2 |
|---|---|---|
| 26 | good | good |
| 2 | bad | bad |
| 1 | good | bad |
| 1 | bad | good |

Table 6.1: Evaluation Results by Two Judges for the RAs of *d5-50*

and can be inconstant. This happens even when the data set is very small, like the *d5-50*. But, the *goodness* measure does confirm that 90% of the RAs are indeed good representative for the underlying feedbacks, and they are coherent with the overall clustering theme.

### 6.5.3 Discussion

We understand that selecting RAs is a very challenging task. The success depends on the quality (in terms of format, grammar, punctuation, spelling, etc.) of the feedbacks, and also relies on some other NLP techniques. In this work, we have proposed a practical, yet effective, approach to select RAs. The experimental results show that our approaches are promising, and the selected RAs are valuable addition to the SD.

However, because of the difficult and complex nature of the problem, we find that our approach has the following major limitations, which should be addressed in future works perhaps using other types of techniques.

- POS tagging quality. Our RA selection is based on the POS tags of the sentences. Therefore, the quality of the POS tagging process has great impact on our final RAs. Note that the POS tagging quality is not only affected by the accuracy of the POS tagger itself, but also the quality of the actual

feedbacks, such as grammar, punctuation, etc.

- Accuracy about the argument orientation. In this work, the orientation of an argument is determined by the orientation of all opinion words that appear in the sentence of the argument. We noticed that there are two major sources that directly contributed to mis-classifying of the orientation. First, we mainly considered adjectives (few particular verbs) as opinion words. However, sometimes other words (such as nouns, verbs) can convey strong ordination information, which will be missed by our approach. Second, even though our approach handled some of the negation words (such as "not", "no") and some of sentimental change sub-clauses (such as "but", "however"), we only did that in a limited extent. Our approach does not cover other complicated cases, which can reverse the overall orientation.

- Handling English idioms or sarcasm. Our approach is unable to handle sarcastic statements, which are often hard for human to understand. Our approach is also lacking the capability to recognize orientations that conveyed by idioms. For example, our approach will consider the sentence "Please get rid of the those language that favor those big enterpriser." as positive because of the word "favor". But this should be considered as negative in terms of the attitude towards the rule, because of the phrase "get rid of". ■

Even though there are some limitations in our *RAPDC* approach, we believe that the RA quality is still acceptable, considering that RAs are only a part of the overall summaritive digest (SD).

## 6.6 Summary

In this chapter, we have introduced our *RAPDC* approach (which stands for RAs selection based on pularity, diversity and CC-coherence) to select RAs for ERFR clusterings. The RAs can present rule-writers some typical arguments so that they can have some sense of the underlying arguments by just looking at the RAs.

We considered *popularity*, *diversity* and *CC-coherence* as three important factors for selecting RAs. The *RAPDC* approach first selects the RAs by only considering the *popularity* and *diversity* factors within a given cluster. Then, steps are taken to ensure that the RAs are coherent across the clusters.

We dealt with trade-offs between *popularity* and *diversity* by dividing the RA quota to address each factor based on a user given threshold. For the *popularity* factor, we performed mini-clustering on those candidate arguments to divide similar arguments into groups. The arguments around "centroid" in the largest argument cluster were selected as popular arguments for that feedback cluster. For the *diversity* factor, we first tried to select those arguments with different sentiments. We then considered content dissimilarity if the sentiment is the same. Additional validation was recommended to reduce inconsistence and to ensure that the RAs are coherent across the entire clustering.

Human efforts were used to evaluate the quality of RAs. Because human judgment is time-consuming and expensive, we conducted small-scale evaluation on some data sets that have been used in Chapter 5. To measure the consistency of human judgement, we calculated the *kappa* for those evaluation. Experimental results showed that the RAs produced by our *RAPDC* approach have good quality, and they can provide useful information about the ERFR to the rule-writers.

Even though the RAs have showed promising results in digesting large ERFR, we believe that the following works can be done to further improve the quality of RAs: (1) studying better ways to deal with trade-offs between *popularity*

and *diversity*, (2) finding ways to automatically validate the *CC-coherence* when considering RAs across the clustering, which may involve sematic analysis, (3) exploiting other important factors and other ways for selecting RAs, (4) utilizing more sophisticated NLP techniques for RA selections, and (5) investigating more subjective ways to systemically evaluate RA quality.

In conclusion, RAs are an important part of the summaritive digest (SD). An SD consists of a clustering, along with SCDs and RAs for each cluster of the clustering, and it can serve as an informative navigation aid to the rule-writers to digest large ERFR efficiently and effectively.

# Chapter 7

# Conclusions and Discussion

As more and larger document repositories become available, the challenges for managing and digesting them efficiently and effectively have become unprecedented. Specially, when increasingly large number of government agencies adopt the e-rulemaking process, the needs for better ways to organize and analyze those large amount of feedbacks become more urgent. This research was primarily motivated by such needs, and was conducted to address them.

In this chapter, we will give a brief summary of this dissertation, and also highlight some major contributions. Based on the results of our work, some future research directions are also described.

## 7.1 Summary

In this dissertation, we have studied the problem of how to effectively digest large document repositories. We considered it for both the generic setting and the special application of e-rulemaking. Specially, we proposed methods to producing informative summaritive digest (SD) for large e-rulemaking feedback repositories (ERFRs). Roughly speaking, the SD is a document clustering, in which each

cluster has succinct cluster descriptions (SCDs) and some representative arguments (RAs). We believe that the SD can be an informative navigation aid for the rule-writers and analysts to manage and digest large amount of feedbacks.

Below, we will give a brief summary of our works based on the chapters.

- In Chapter 1, we gave the motivations for this research, and set the research goal for this dissertation. We also briefly highlighted some of the related approaches that addressed similar problems as ours.

- In Chapter 2, we presented some preliminaries on the techniques and terminologies that are used throughout this dissertation. These include e-rulemaking (electronic rulemaking), document clustering, document summarization and some related works in those areas. In addition, we also gave brief introduction on part-of-speech (POS) tagging and WordNet.

- In Chapter 3, we argued that stakeholders (S), issues (I) and opinions (O) are three very important aspects for organizing ERFRs. We proposed practical methods to identify those major stakeholders, important issues and relevant opinions. For identifying S and I terms, we applied association mining techniques on the nouns and noun phrases extracted from the feedbacks based on the POS tags, although we also suggested to incorporate user feedbacks to finalize the S and I lists. Since we were only interested in those opinions that have been expressed by major stakeholders on some of the those important issues, we used simple syntax parsing approach (and some cue phrases) to identify opinions based on the appearance of S and/or I. We also utilized WordNet for determining the orientation of opinion terms.

- In Chapter 4, we studied the problem of cluster descriptions (CDs) extensively for any given document clustering, regardless of the clustering algorithm used to produce the clusters. We discussed and formalized how to

interpret the CDs, what measures should be used to find good CDs, and how to find good CDs. We proposed the CDD surrogate measure, which is based on three factors of *coverage*, *disjointness*, and *diversity*. We also introduced a *CD-based classification* approach to systematically evaluate CD quality. In addition, we presented a layer-based replacement search method called PagodaCD and a greedy forward search method called *CumulativeCD* for constructing CDs. At the end of this chapter, we also reported our initial efforts on using genetic algorithm (GA) to construct CDs.

- In Chapter 5, we introduced our *OIS-based* approach to perform clustering and construct SCDs simultaneously for large ERFR.

  For the clustering process, we used our novel *active feature selection* (AFS) and *adaptive similarity measure* (ASM) approaches. The AFS approach utilizes the proposed rule and general-topic document collection as *background knowledge* for helping the feature selection. In the ASM approach, the feature weights can be adjusted to meet the different needs of users.

  For the SCD construction, we provided more quantitative information to enhance the CD that we discussed in chapter 4. In addition to the CDD measure, we also tried to use the CU measure as another surrogate measure for searching quality SCDs. We proposed an enhanced version of the PagodaCD algorithm, called PagodaCD+, which can utilize any improvement measures and can also obtain additional statistical information for the SCDs.

- In Chapter 6, we introduced our *RAPDC* approach to select RAs for ERFR clusterings. The approach considered *popularity*, *diversity* and *CC-coherence* as three important factors for selecting RAs. We dealt with the trade-offs between *popularity* and *diversity* by dividing the RA quota to address each factor based on a user given threshold. Additional checkup was also recom-

mended to ensure the RAs are coherent across clusters. ■

Collectively, those chapters presented a complete picture of our approach to producing SD for digesting large ERFR. Experimental evaluations have been conducted for those proposed approaches, and the results have shown that our approaches are very promising and effective. However, limitations were also observed in some of the approaches, which have been suggested as potential future works.

In this dissertation, we have focused on an important IT component for effectively and efficiently digesting large ERFRs. However, the IT approach alone may not be able to solve the problem completely. Moreover, there are also a variety of other issues that need to be addressed (such as social, political, and other technical issues) to make our approaches more useable and more effective.

## 7.2 Contributions

Having achieved the initial research goal, which is to produce informative SDs for large ERFRs, is a big accomplishment. Major contributions made in this dissertation can be summarized as the following:

- We conducted extensive studies on cluster description (CDs), which is based on a small set of terms. We discussed and formalized (i) how to interpret CDs, (ii) how to measure the quality of CDs (e.g. CDD measures,), and (iii) how to produce high-quality CDs (e.g. PagodaCD algorithm).

- We proposed the *CD-based classification* approach to approximate how the CD should be "interpreted". This approach provides an alternative way to automatically measure CDs. That is, the difference between the classification result and the original clusters was used to measure the CD quality.

- We introduced a group of search algorithms for constructing SCDs, such as the PagodaCD, PagodaCD+ and *CumulativeCD* algorithms. PagodaCD+ is an improved version of PagodaCD, and can work with any improvement measures. Both of these two pagoda methods divide the search process into multiple major steps, working in a layered manner. Those algorithms have been used to find good SCD terms, and they are efficient and can produce SCDs with monotone quality behavior.

- We studied how to perform clustering and construct SCDs simultaneously for given ERFRs, so that both the clustering quality and the SCD quality are addressed at the same time. We also proposed to use the *active feature selection* (AFS) and *adaptive similarity measure* (ASM) techniques in the clustering process.

- We proposed the *OIS-based* approach to produce SD for given ERFR. The approach utilizes three important aspects of ERFRs (i.e. stakeholders, issues and opinions) throughout the process.

- We introduced the *RAPDC* approach to select RAs for ERFR clusterings. The approach takes consideration of three factors of *popularity*, *diversity* and *CC-coherence* when selecting RAs. ■

We believe that those contributions are not only beneficial for managing large ERFRs, but also for other document repositories.

Before concluding this section, it is worth mentioning that the SCD alone can also be very useful for many emerging applications. One of them is to treat the terms in SCD as *tags*. Tags are typically used to generate internet taxonomies for online resources, often called folksonomy[128]. The folksonomic tagging is intended to make a body of information increasingly easy to search, discover, and navigate over time. Two widely cited examples of websites using folksonomic

tagging are Flickr[1] and del.icio.us[2]. Another possibility is using SCDs to build one-class classifiers [78, 16], so that they can be used to incrementally adding new documents to large document repositories.

## 7.3 Future Work

While the work in this dissertation has addressed many problems of digesting large ERFRs, it could only do so to a limited depth, and in a limited point of view. We believe that this dissertation has laid the foundation for a wide variety of potential research and applications.

In this section, we will give two major research directions. First, we will outline some natural improvements to the current work. Then, we will exploit the idea of using linked SDs produced from different time snapshots to find "drifting concept" over time, which can be very useful for the applications of digital government and digital archiving.

### 7.3.1 Improvement to Current Works

While our work has showed promising results for organizing ERFRs, we believe that the SD quality can be further improved from the following additional research:

- In the CDD measure, we treated three factors of *coverage*, *disjointness* and *diversity* as equally important. However, their importance may not be always the same for different domains and applications. Therefor, those factors may need to have different weights depending on the situation.

- When constructing CDs, we only considered those terms that appeared in the target documents. One could consider terms that do not occur in the tar-

[1] http://www.flickr.com
[2] http://wwww.del.icio.us

get documents. Such efforts could utilize some domain-specific ontology, or consider synonyms or other taxonomy to make the CD more human friendly.

- We have treated *stakeholders*, *issues* and *opinions* as important aspects for managing large ERFR. There could be other important factors that need to be further exploited.

- The *RAPDC* approach considered *popularity*, *diversity* and *CC-coherence* factors when selecting RAs for ERFR clustering. However, there are still parameters that need to be estimated, and the *CC-coherence* factor can not be enforced automatically. Therefore, there is a need to automatically incorporate all those factors and balance the weights among them.

- While the SD quality will be ultimately judged by user's happiness, it is desirable to find some objective ways/measures to automate the evaluation process, because human judgment is time-consuming, expensive and maybe inconsistent. We believe that such efforts will also bring benefits to other type of document related evaluations. ■

### 7.3.2 Linking SDs for Document Archive History

Many government agencies, private corporations, not-for-profit organizations, and even private citizens are now concerned with preserving their own digital information assets. Therefore, digital archiving has rapidly become a critical issue in recent years, since more and more valuable content is "born digital" and must be managed, preserved, and used in digital form [93].

One of the challenging problems is logical repurposing [94]. For a given digital archive, users not only want to know the topics at a specific time, but also the major changes and major invariants over time. We believe that SDs can be used for this purpose. We could construct SDs at different archive snapshots, and integrate

those SDs together so that they can give users a high-level sense about what have changed over time.

Digital archiving is still a very active research area. However, we have not seen any work to produce SD, especially threaded SDs linked by time. We believe that the following important questions need to be addressed:

- How to construct SD for document archives at a particular time point?
- How can we integrate the SDs at different time points to form a global picture about the archive?

**Construct SD at a Particular Timestamp.** As we know, the most popular archiving approach is to store a sequence of deltas. With this archiving approach, commonly called the *diff approach*, we may need to undo or redo a large number of changes to get a full copy of the contents for a particular time point. This recovery process may also need significant reasoning with the deltas [13]. Furthermore, some archiving approaches may also consider the redundancy, security and other issues during archiving, which also need to be considered. So it is still a challenging work to construct SD for document archives at a particular time point.

**Construct Linked SDs Over Times.** We need to link those SDs at different timestamps together so that the threaded SDs can reflect the major topic changes over time.

Two important questions that need to be addressed are:

- In what order should we construct SDs at different timestamps? The choice of either "forward" or "backward" will depend on the archiving method.
- How to integrate those SDs to reflect the topic changes over time? Since the archive usually evolves with time, the SDs should be able to reflect these changes. How to reconstruct a new SD based on existing SDs or refine those existing SDs is still an open question.

As a side remark, the contents of digital archives may include text, electronic documents, databases, images, sound, video and other object types [93]. Therefore, there is another layer of challenge, which is to consider those heterogeneous content types.

# Appendix A: Feedback Examples

## A.1 Examples from Dataset DoA-NOP [3]

### 0. The Proposed NOP Rule

DEPARTMENT OF AGRICULTURE

Agricultural Marketing Service

7 CFR Part 205

[Docket Number: TMD-94-00-2] RIN: 0581-AA40

National Organic Program

AGENCY: Agricultural Marketing Service, USDA.

ACTION: Proposed rule.

SUMMARY: The Agricultural Marketing Service (AMS) is seeking comments on a proposal to establish a National Organic Program (NOP or program). The program is proposed under the Organic Foods Production Act of 1990 (OFPA or Act), as amended, which requires the establishment of national standards governing the marketing of certain agricultural products as organically produced to facilitate commerce in fresh and processed food that is organically produced and to assure consumers that such products meet consistent standards. This program would establish national standards for the organic production and han-

[3]Even though the data sets used in this thesis are publicly available, we still omitted the submitter's name from the following examples, if there is one, for privacy reason.

dling of agricultural products, which would include a National List of synthetic substances approved for use in the production and handling of organically produced products. It also would establish an accreditation program for State officials and private persons who want to be accredited to certify farm, wild crop harvesting, and handling operations that comply with the program's requirements, and a certification program for farm, wild crop harvesting, and handling operations that want to be certified as meeting the program's requirements. The program additionally would include labeling requirements for organic products and products containing organic ingredients, and enforcement provisions. Further, the proposed rule provides for the approval of State organic programs and the importation into the United States of organic agricultural products from foreign programs determined to have equivalent requirements.

**1. Feedback Example 1 About the NOP Rule**

I strongly oppose the proposed rules for organics as currently written. As a business person, an educator, and a cancer patient, I find the efforts of USDA laudable but seriously lacking in several key areas. Specifically: 1. States, such as Oregon, must be allowed to require out-of-state farm and food handling operations to comply with more stringent regulations beyond the national standards. 2. Organics may NOT include: a. genetic engineering nor food irradiation b. the use of sewage fertilizer or "iosolids" c. non-organic seed and planting stock d. the use of vaccinations, antibiotics, or other drugs e. the application of synthetic or other "dormant" substances on crops f. use of synthetic rodent and insect poisons. Clearly, the rules are intended to satisfy the industry and agrobusiness rather than the consumer. Therefore, I

encourage a strengthening of the rules as indicated above. Until such time, I will continue to buy only goods certified by Oregon Tilth, which are considerably more strict - and healthier - than those that would be available under the proposed guidelines. Respectfully submitted

**2. Feedback Example 2 About the NOP Rule**

I am outraged as a consumer that irradiated foods, gentically engineered foods, and foods grown on lands fertilized with sewage sludge are included in the USDA's proposed rules for organic foods. These practices have never been a part of organic food agriculture. Inclusion of these types of foods is irresponsible. I hope this rule serves one purpose–to convince people to grow their own healthy organic foods and not be victimized by federal macromanagement strategies that pander to large-scale agribusinesses. This rule is indicative of how mainstream macromanagement policies continue to corrupt what is a God given right for all who inhabit Earth–access to healthy foods. I enjoy the fruits of a 38 acre burgeoning farm in the Blue Ridge mountains of Virginia, but for those who depend on responsible agricultural practices to provide their families with healthy foods, all I can say is "God help you because the government will not."

**3. Feedback Example 3 About the NOP Rule**

I am a consumer of organic food products and wish that methods such as sewage sludge, irradiation, and biotechnology be banned from the production of organic foods. The purpose of "organic" is to produce food that is free from chemicals, and have a richer nutrient content due to farming techniques. Thank you.

## A.2 Examples from EPA-CWA Data Set

### 1. Feedbacks Example About the CWA Rule

Note: In the following feedback, all the "‘" were appeared as "?". We made the change for readability.

> I‘m urging the EPA and Army Corps not to proceed with a rulemaking to redefine Waters of the United States . I also urge the agencies to withdraw the guidance document immediately. Instead, the agencies should focus on implementing a narrow interpretation of the Supreme Court‘s decision. The agencies should also clarify to their field offices that the only waterways that should be excluded from protection are those directly addressed by the Supreme Court‘s ruling. The Clean Water Act was enacted to "restore the physical, chemical and biological integrity of our nation‘s waters." Unfortunately, your actions threaten to do the opposite, turning back the 30 years of progress made under the Clean Water Act and leading to significantly more flooding, pollution and accelerated loss of wildlife habitat. I have worked with farmers to decrease toxic runoff and improve water quality here in California, and have come to the conclusion that our state is in dire need of EPA protection for our waterways. With wetlands disappearing and a huge number of endangered species, Californians support the Clean Water Act and the progress that has been made. Please reverse your decision to exclude important waterways from protection! Sincerely

# Appendix B: Additional Results

**B.1** Some of the most frequent terms that appeared in data set DoA-NOP-0, also some stakeholders and issues that selected from the candidate list. Note that words are in their root format.

### B.1.1 The 45 most frequent terms

usda wai food sewag sludg wast rule organ product practic farmer propos consum industri irradi fertil definit state standard health pesticid farm peopl nation board engin crop materi radiat biosolid synthet chemic ingredi choic process govern anim substanc label certif produc year term market antibiot

### B.1.2 Stakeholders selected from the candidate list

citizen consum crop famili farmer govern health industri peopl plant program state usda

### B.1.3 Issues selected from the candidate list

anim antibiot biotechnolog certif chemic contamin engin fertil food irradi issu label metal natur organ product regul sewag sludg standard wast

**B.2** Some candidate arguments identified for data set *d5-1k*. Note that the arguments have the format [**documentIndex, sentenceIndex, orientation, sentence**], where both indexes started from 0. Orientations are p, n and u, which means positive, negative and unknown, respectively. Note that this format is different than the format of RAs.

0,1,n,Genetically engineered, irradiated, and sludge-grown products are by definition NOT organic.

0,4,n,Irradiated or "nuked" products have been altered from their natural state and are therefore not organic.

2,0,n,I am concerned about the potential inclusion of genetically modified organisms, food irradiation, the use of antibiotics in livestock and dairy production and the use of sewage sludge as fertilizer.

2,1,u,Please follow the standards that the small organic farmers have been using.

3,1,p,I applaud your providing definition and enforcement to "organic" claims but I wish with all earnestness you would restrict the biotechnology, irradiation and human sewage as unacceptable under the definition of organic.

...

25,2,u,My only change to this proposed standard would be the removal of any requirement that would prohibit other certifying organizations from enforcing stricter standards.

26,0,p,I support organic industry standards.

26,1,n,The parts of the Preamble to the Standards which I object to relate to the possible inclusion of irradiation, antibiotics used in the

production of livestock and dairy, the use of sewage sludge as fertilizer and genetically engineered organisms.

26,2,p,I support prohibiting each of the above.

...

32,23,n,The organic food industry has struggled against tremendous odds to survive thus far and we damned sure don't want to see regulation favor conglomerate demands to water down standards that have been a long, hard fight to win.

33,0,p,I highly applaud the federal government with finally coming out with national organic food standards.

37,1,p,I was pleased to hear that the USDA is taking steps to further legitimize the organic food industry.

37,2,u,I am a consumer of organic foods as well as a grower of my own organic vegetables.

37,3,n,I am very concerned, however, about the USDA overlooking food which is irradiated and bio-engineered.

38,0,p,I applaud the effort today to provide uniform rules for organic produce and meat products.

38,1,n,My major area of concern is the failure to include genetically altered substances, irradiated products and human waste compost from sewerage facilities in the banned substances.

...

**B.3** Selected RAs for data set *d5-50*, which is used in Figure 6.5. Note that the RAs have the format [**clusterIndex, orientation, documentIndex, sentenceIndex, argument**], where indexes started from 0.

### B.3.1 RAs for Cluster 0

0,n,17,0,I do not want any food item to be labeled with your 'organic' seal if it has been produced with the use of sludge, biotechnology or irradiation unless that food product says so clearly on the label.

0,n,18,0,I am very concerned that the USDA is merely interested in allowing industrialized food engineering companies a means to label their materials as organic– pure, clean, and real.

0,n,40,7,This is slanted to large organization and the clear agenda is to consolidate there power and squeeze out the grass roots organizations, giving them a monopoly.

0,n,47,0,I entirely oppose the USDA's effort to dilute the 1990 Organic products definitions.

0,u,1,20,If so, I suggest you correct the wording in 205.103 to CLEARLY state your intent.

### B.3.2 RAs for Cluster 0.0

0,n,5,0,I do not want your ' organic ' label to be put on any produce that has been produced with the use of sludge, biotechnology or irradiation unless that produce is clearly labeled as having been produced with the use of these techniques.

0,u,1,0,I feel strongly about having gen. engeneered and Iradataded prouducts under the same rules for organic products.

0,u,1,1,Labeling these Products will help the cunsumer make and informed chose to have a true organic food stuff.

0,u,5,1,I also want any 'organic' labeled food to say clearly somewhere on the label whether it has had a 'veggie wash' containing iodine.

0,u,5,4,Please do not let the big food companies dilute the current organic certification rules just so that they can muscle in on the BIG ( $ Four Billion ) organic food market with minimum disruption of their current non-organic methods.

### B.3.3 RAs for Cluster 0.1

1,n,0,8,However, I strongly disagree with a number of OCIA's standards, feeling that they are TOO LAX.

1,n,11,1,The meaning of organic means that no pesticides or unnatural materials such as sewer sludge, irradiation, bio-engineering, inorganic seeds and seedlings, ground sheep brains for cattle feed, antibiotics, coatings for fruit, botanical pesticides and many more items in a long list of non-organic practices will be available to farmers to gain the appellation of "Organic".

1,n,12,7,This is slanted to large organization and the clear agenda is to consolidate there power and squeeze out the grass roots organizations, giving them a monopoly.

1,n,4,15,To not take a powerful stand against these means and include them as part of your definition will certainly make the term "organic" laughable indeed.

1,u,0,3,The term has been around a very long time, and no regulation is going to change its use.

### B.3.4 RAs for Cluster 1

1,n,36,6,I do feel, however, that it is a good test to begin with.

1,p,0,9,You can't treat the word "organic" as a term of art and redefine it solely to satisfy pressures put upon your agency by mainstream non-organic food producers, distributors and lobbyists.

1,p,26,0,I support organic industry standards.

1,p,32,3,If the government goes with less strict standards than what independent organic certification bodies consider acceptable, consumers will not be able to distinguish between products that meet the lower level of standards and those that meet a higher standard.

1,u,34,17,I will offer more comments as I finish reading all of the regulation and discussing them with my customers, but I feel compelled to give you this initial reaction from me and the customers I represent who are committed organic consumers.

### B.3.5 RAs for Cluster 1.0

0,p,31,0,I buy organic foods for my family, and I want to be sure that national regulation will mean the highest possible standards.

0,p,32,2,I strongly feel that these practices are neither safe, nor healthy and are in direct opposition to the letter and intent of the term organic.

0,p,7,1,Consumers have the right to know that in making the selection of organic foods we are protected from all the inciduous processes which are used blatantly to mass produce the products we avoid and for which we pay dearly in cash/time to avoid.

0,u,31,3,I am strongly opposed to these provisions, which are inconsistent with natural organic agriculture.

0,u,9,0,I believe it is imperative that food products which are advertised as "organically produced" are in fact produced in a manner that keeps

them free from manmade compounds and processes which have been shown, or have the potential to produce adverse health effects.

### B.3.6 RAs for Cluster 1.1

1,p,13,1,I support keeping the use of sewage sludge out of the final regulations.

1,p,21,3,Even the idea of a completely sterlized food supply may not be acceptable, considering that benefitial bacteria inhabitng the gastrointestinal tract are helpful in the human digestive process.

1,u,15,38,What we REALLY need in this country is education for all that isn't polluted by Industry as our government is.

1,u,19,1,Consumers expect organic to mean, produced naturally.

1,u,23,1,Since you published your proposed regulations of organic foods I have been asking my customers what they thought of some of the main points you have asked for comment on.

# Bibliography

[1] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM SIGMOD*, pages 94–105, 1998.

[2] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 26–28 1993.

[3] Rakesh Agrawal and John C. Shafer. Parallel mining of association rules. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):962–969, 1996.

[4] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 12–15 1994.

[5] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet project. In Christian Boitet and Pete Whitelock, editors, *Pro-*

*ceedings of the 36 Annual Meeting of the ACL and 17th Int. Conf. on CL*, pages 86–90, San Francisco, CA, 1998. Morgan Kaufmanns.

[6] Sugato Basu, Mikhail Bilenko, and Raymond J. Mooney. A probabilistic framework for semi-supervised clustering. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 59–68, New York, NY, USA, 2004. ACM Press.

[7] Florian Beil, Martin Ester, and Xiaowei Xu. Frequent term-based text clustering. In *Proc. 8th Int. Conf. on KDD*, 2002.

[8] Michael W. Berry. *Survey of Text Mining: Clustering, Classification, and Retrieval*. Springer-Verlag, New York, 2003.

[9] Steven Bethard, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. Automatic extraction of opinion propositions and their holders. AAAI Spring Symposium on AAAI-EAAT 2004.

[10] Gautam Biswas, Jerry B. Weinberg, and Doug H. Fisher. ITERATE: A conceptual clustering algorithm for data mining. *IEEE Transactions on Systems, Man, Cybernetics*, 28C(2):219–230, 1998.

[11] Didier Bourigault and Christian Jacquemin. Term extraction + term clustering: an integrated platform for computer-aided terminology. In *Proc. of the 9th conf. on EACL*, pages 15–22, Morristown, NJ, 1999.

[12] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[13] Peter Buneman, Sanjeev Khanna, Keishi Tajima, and Wang-Chiew Tan. Archiving scientific data. *ACM Trans. Database Syst.*, 29(1):2–42, 2004.

[14] Carnegie Mellon University E-Rulemaking Project. http://hartford.lti.cs.cmu.edu/erulemaking.

[15] Soumen Chakrabarti. Data mining for hypertext: A tutorial survey. *SIGKDD Explorations Newsletter, ACM*, 1, 2000.

[16] Lijun Chen and Guozhu Dong. Masquerader Detection Using OCLEP: One-Class Classification Using Length Statistics of Emerging Patterns. International Workshop on INformation Processing over Evolving Networks (WINPEN), 2006. (Proceedings published by IEEE.).

[17] Lijun Chen and Guozhu Dong. Succinct and informative cluster descriptions for document repositories. In *International Conference on Web-Age Information Management (WAIM)*, pages 109–121, 2006.

[18] Philipp Cimiano, Andreas Hotho, and Steffen Staab. Clustering Concept Hierarchies from Text, 2004.

[19] Philipp Cimiano, Andreas Hotho, and Steffen Staab. Comparing Conceptual, Divisive and Agglomerative Clustering for Learning Taxonomies from Text, 2004.

[20] Cary Coglianese. E-Rulemaking: Information Technology and Regulatory Policy. Regulatory Policy Program Report No. RPP-05 (2004).

[21] Padraig Cunningham and John Carney. Diversity versus quality in classification ensembles based on feat ure selection. In *European Conference on Machine Learning*, pages 109–116, 2000.

[22] Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *The 12th Int. World Wide Web Conf. (WWW2003)*, 2003.

[23] Inderjit S. Dhillon, James Fan, and Yuqiang Guan. Efficient clustering of very large document collections. In R. Grossman, G. Kamath, and

R. Naburu, editors, *Data Mining for Scientific and Engineering Applications*. Kluwer Academic Publishers, 2001.

[24] Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1/2):143–175, 2001.

[25] DMOZ: Open Directory Project. http://dmoz.org.

[26] Guozhu Dong and Jinyan Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proc. of the 5th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 1999.

[27] Guozhu Dong, Xiuzhen Zhang, Limsoon Wong, and Jinyan Li. CAEP: Classification by aggregating emerging patterns. In *Discovery Science*, pages 30–42, 1999.

[28] DUC. Document understand conferences. http://duc.nist.gov.

[29] Agoston E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. SpringerVerlag, 2003.

[30] Barbara Di Eugenio and Michael Glass. The kappa statistic: a second look. *Comput. Linguist.*, 30(1):95–101, 2004.

[31] Ronen Feldman and Haym Hirsh. Mining associations in text in the presence of background knowledge. In *Knowledge Discovery and Data Mining*, pages 343–346, 1996.

[32] Christiane Fellbaum. *WordNet An Electronic Lexical Database*. The MIT Press, London, 1998.

[33] Douglas H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.

[34] Jane E. Fountain. Prospects for improving the regulatory process using e-rulemaking. *Communications of the ACM*, 46(1):43–44, 2003.

[35] William B. Frakes and Ricardo Baeza-Yates. *Information Retrieval Data Structures and Algorithms*. Prentice Hall, Englewood Cliffs, NJ, 1992.

[36] Alex A. Freitas. A survey of evolutionary algorithms for data mining and knowledge discovery. In *Advances in evolutionary computing: theory and applications*, pages 819–845, New York, NY, USA, 2003. Springer-Verlag New York, Inc.

[37] Benjamin C.M. Fung, Ke Wang, and Martin Ester. Hierarchical document clustering using frequent itemsets. In *Proceedings of SIAM International Conference on Data Mining*, 2003.

[38] Mark A. Gluck and James E. Corter. Information, uncertainty, and the utility of categories. In *Proc of the Seventh Annual Conference of the Cognitive Science Society*, 1985.

[39] Jade Goldstein, Mark Kantrowitz, Vibhu O. Mittal, and Jaime G. Carbonell. Summarizing text documents: Sentence selection and evaluation metrics. In *R & D in Information Retrieval*, pages 121–128, 1999.

[40] Gene H. Golub and Charles F. Van Loan. Matrix Computations. John Hapkins Univ. Press, Baltimore, Maryland, 3rd edition, 1996.

[41] A.D. Gordon. *Classification, 2nd ed.* Chapman & Hall, 1999.

[42] Yuqiang Guan. Large-Scale Clustering: Algorithms and Applications in Text Mining and Bioinformatics. PhD dissertation proposal, 2003.

[43] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. CURE: An efficient clustering algorithm for large databases. In *SIGMOD*, pages 73–84, 1998.

[44] Eui-Hong Han, George Karypis, Vipin Kumar, and Bamshad Mobasher. Clustering based on association rule hypergraphs. In *Research Issues on Data Mining and Knowledge Discovery*, 1997.

[45] J. Han and Y. Fu. Exploration of the power of attribute-oriented induction in data mining. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 399–421. AAAI/MIT Press, 1996.

[46] Jiawei Han, Yandong Cai, and Nick Cercone. Knowledge discovery in databases: An attribute-oriented approach. In Li-Yan Yuan, editor, *Proceedings of the 18th Int. Conf. on VLDB*, pages 547–559, San Francisco, USA, 1992. Morgan Kaufmann Publishers.

[47] Jiawei Han and Yongjian Fu. Exploration of the power of attribute-oriented induction in data mining. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhr Smyth, and Ramasamy Uthurusamy, editors, *Advances in KDD*. AIII/MIT Press, 1996.

[48] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Mogan Kaufmann, San Francisco, CA, USA, 2001.

[49] Harvard Unviersity eRulemaking Resource Website. http://www.ksg.harvard.edu/cbg/rpp/erulemaking.

[50] Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the 8th conf. on EACL*, pages 174–181, Morristown, NJ, USA, 1997.

[51] Simon Haykin. *Neural Networks A Comprehensive Foundation*. Prentice-Hall, 1999.

[52] Marti A. Hearst, David R. Karger, and Jan O. Pedersen. Scatter/gather as a tool for the navigation of retrieval results. In *Working Notes AAAI Fall Symp. AI Applications in Knowledge Navigation*, 1995.

[53] Andreas Hotho, Steffen Staab, and Gerd Stumme. Wordnet improves text document clustering. *in submitted*, 2003.

[54] Andreas Hotho and Gerd Stumme. Conceptual clustering of text clusters. In *Proceedings of FGML Workshop*, pages 37–45, 2002.

[55] Eduard Hovy and Chin yew Lin. Automated text summarization in SUMMARIST. In Mani and Maybury, 18–24., 1997.

[56] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *KDD '04: Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, New York, NY, USA, 2004. ACM Press.

[57] Yifen Huang and Tom M. Mitchell. Text clustering with extended user feedback. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 413–420, New York, NY, USA, 2006. ACM Press.

[58] C. Jacquemin and D. et. Term extraction and automatic indexing, 2000.

[59] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[60] Jamie Callan. erulemaking testbed. http://hartford.lti.cs.cmu.edu/erulemaking/data.html.

[61] Xiang Ji and Wei Xu. Document clustering with prior knowledge. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 405–412, New York, NY, USA, 2006. ACM Press.

[62] Lou Jost. Entropy and diversity. OIKOS (is a journal of ecology). (Accepted 3 January 2006).

[63] George Karypis. Cluto: A clustering toolkit (release 2.1.1). http://www-users.cs.umn.edu/~karypis/cluto/.

[64] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, Inc., NY, 1990.

[65] YongSeog Kim, W. Nick Street, and Filippo Menczer. Feature Selection in Unsupervised Learning via Evolutionary Search. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000.

[66] L. Kuncheva, C. Whitaker, C. Shipp, and R. Duin. Limits on the majority vote accuracy in classier fusion.

[67] Namhee Kwon, Stuart W. Shulman, and Eduard Hovy. Multidimensional text analysis for erulemaking. In *dg.o '06: Proceedings of the 2006 international conference on Digital government research*, pages 157–166, New York, NY, USA, 2006. ACM Press.

[68] TBjornar Larsen and Chinatsu Aone. Fast and effective text mining using linear-time document clustering. In *Proc. of KDD 99*, pages 16–22. ACM, 1999.

[69] Gloria T. Lau. *A comparative analysis framework for semi-structured documents, with applications to government regulations.* PhD thesis, Stanford University, 2004. Adviser-Kincho H. Law.

[70] Gloria T. Lau, Shawn Kerrigan, Kincho H. Law, and Gio Wiederhold. An e-government information architecture for regulation analysis and compliance assistance. In *ICEC '04: Proc. of the 6th int. conf. on E-commerce*, pages 461–470, New York, 2004. ACM Press.

[71] Gloria T. Lau, Kincho H. Law, and Gio Wiederhold. A relatedness analysis tool for comparing drafted regulations and associated public comments. dg.o 2005.

[72] Gloria T. Lau, Kincho H. Law, and Gio Wiederhold. Similarity analysis on government regulations. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 711–716, New York, 2003. ACM Press.

[73] Brian Lent, Arun N. Swami, and Jennifer Widom. Clustering association rules. In *ICDE*, pages 220–231, 1997.

[74] Todd A. Letsche and Michael W. Berry. Large-scale information retrieval with latent semantic indexing. *Info. Sciences*, 100(1-4):105–137, 1997.

[75] David D. Lewis. Reuters-21578 text categorixation test collection. http://www.daviddlewis.com/resources/ testcollections/reuters21578/, 1997.

[76] Wenmin Li, Jiawei Han, and Jian Pei. CMAR: Accurate and efficient classification based on multiple class-association rules. In *ICDM*, pages 369–376, 2001.

[77] Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In *KDD '98*, pages 80–86, 1998.

[78] Bing Liu, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. Text classification by labeling words. In *Proceedings of Nineteenth National Conference on Artificial Intelligence*, pages 425–430, Menlo Park, California, USA, 2004. The AAAI Press.

[79] Shuang Liu, Fang Liu, Clement Yu, and Weiyi Meng. An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In *SIGIR '04: Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 266–272, New York, NY, USA, 2004. ACM Press.

[80] Ying Liu, Shamkant B. Navathe, Jorge Civera, Venu Dasigi, Ashwin Ram, Brian J. Ciliax, and Ray Dingledine. Text mining biomedical literature for discovering gene-to-gene relationships: A comparative study of algorithms. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 2(1):62–76, 2005.

[81] Inderjeet Mani and Eric Bloedorn. Summarizing similarities and differences among related documents. *Inf. Retr.*, 1(1-2):35–67, 1999.

[82] Chris Manning and Hinrich Schtze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, MA, 1999.

[83] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. In *Computational Linguistics*, volume 19, number 2, pp313-330.

[84] Mark T. Maybury and Inderjeet Mani. Automatic summarization. Tutorial Notes on ACL/EACL 2001.

[85] Kathleen McKeown, Judith Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. Towards multidocument summarization by reformulation: Progress and prospects. In *AAAI/IAAI*, pages 453–460, 1999.

[86] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 171–180, New York, NY, USA, 2007. ACM Press.

[87] George A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[88] Melanie Mitchell and Charles E. Taylor. Evolutionary computation: An overview. *Annual Review of Ecology and Systematics*, 20:593–616, 1999.

[89] Raymond J. Mooney and Razvan Bunescu. Mining knowledge from text using information extraction. SIGKDD *explorations (special issue on text mining and natural language processing), 7, 1 (2005), pp. 3-10.*

[90] Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. Mining product reputations on the web, 2002.

[91] M. Morita, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Unsupervised feature selection using multi-objective genetic algorithms for handwritten word recognition. In *ICDAR '03: Proceedings of the Seventh International Conference on Document Analysis and Recognition*, page 666, Washington, DC, USA, 2003. IEEE Computer Society.

[92] S. Naranan and V. K. Balasubrahmanyan. Models for power law relations in linguistics and information science. *Journal of Quantitative Linguistics*, 5(1-2):35–61, 1998.

[93] NSF. Digital Archiving and Long-Term Preservation (DIGARCH) Program Solicitation, NSF 04-592, 2004.

[94] NSF and LoC. IT'S ABOUT TIME: Research Challenges in Digital Archiving and Long-term Preservation, *workshop report*. 2003.

[95] Chris D. Paice. Another stemmer. *ACM SIGIR Forum*, 24(3):56–61, 1990.

[96] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, Philadelphia, July 2002. ACL.

[97] Jian Pei, Jiawei Han, and Runying Mao. CLOSET: An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

[98] E. C. Pielou. Shannon's Formula as a Measure of Specific Diversity: Its Use and Misuse, he American Naturalist, Vol. 100, No. 914 (Sep. - Oct., 1966), pp. 463-465.

[99] Martin Porter. An algorithm for suffix stripping, program: Automated library and information systems, 14 (3), pp. 130-137, 1980.

[100] D. Radev, S. Teufel, H. Saggion, W. Lam, J. Blitzer, H. Qi, A. elebi, D. Liu, and E. Drabek. Evaluation challenges in large-scale document summarization, 2003.

[101] Dragomir R. Radev, Hongyan Jing, Magorzata Stys, and Daniel Tam. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40(6):919–938, 2004.

[102] Dragomir R. Radev and Daniel Tam. Summarization evaluation using relative utility. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 508–511, New York, NY, USA, 2003. ACM Press.

[103] Regulations.gov. http://www.regulations.gov.

[104] Dmitri Roussinov and Marshall Ramsey. Information forage through adaptive visualization. In *Proceedings of the third ACM conference on Digital libraries*, pages 303–304, 1998.

[105] Gerard Salton, James Allan, Chris Buckley, and Amit Singhal. Automatic analysis, theme generation, and summarization of machine-readable texts. *Science*, 264(3):1421–1426, 1994.

[106] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[107] R. Shapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[108] Stuart Shulman, Eduard Hovy, Jamie Callan, and Stephen Zavestoski. Language processing technologies for electronic rulemaking: A project highlight. dg.o 2005.

[109] Stuart W. Shulman. eRulemaking: Issues in Current Research & Practice. Prepared for the special eGov issue of the International Journal of Public Administration. 2003.

[110] Stuart W. Shulman. Technological Responses to Mass E-Mail Campaigns in U.S. Regulatory Rulemaking. Public Lecture at the Oxford Internet Institute (Draft). 2005.

[111] Stuart W. Shulman. The Internet Still Might (But Probably Wont) Change Everything. Accepted for publication in I/S Journal. 2004.

[112] Stuart W. Shulman and Mack C. Shelley. Give the people what they want: Research notes from the hunt for better electronic rulemaking. dg.o 2005.

[113] Noam Slonim and Naftali Tishby. Document clustering using word clusters via the information bottleneck method. In *SIGIR '00: Proc.s of the 23rd annual int. SIGIR conf.*, pages 208–215, New York, 2000. ACM Press.

[114] Michael Steinbach, Levent Ertoz, and Vipin Kumar. Challenges of Clustering High Dimensional Data. *new vistas in statistical physics – applications in econophysics, bioinformatics, and pattern recognition*. 2003.

[115] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. In *Proceedings of KDD Workshop on Text Mining*, 2000.

[116] Alexander Strehl. *Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining*. PhD thesis, The University of Texas at Austin, May 2002.

[117] N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[118] Alvin Toffler. *Introduction to Modern Information Retrieval*. Random House, New YorK, NY, USA, 1970.

[119] TREC. Text retrieval conference (trec). http://trec.nist.gov.

[120] Pucktada Treeratpituk and Jamie Callan. Automatically labeling hierarchical clusters. In *dg.o '06: Proceedings of the 2006 international conference*

*on Digital government research*, pages 167–176, New York, NY, USA, 2006. ACM Press.

[121] Peter D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, pages 417–424, 2002.

[122] University of Pittsburgh eRulemaking Research Group. http://erulemaking.ucsur.pitt.edu.

[123] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.

[124] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schroedl. Constrained k-means clustering with background knowledge. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[125] Ke Wang, Chu Xu, and Bing Liu. Clustering transactions using large items. In *CIKM '99: Proceedings of the eighth international conference on Information and knowledge management*, pages 483–490, New York, NY, USA, 1999. ACM Press.

[126] Janyce Wiebe. Learning subjective adjectives from corpora. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 735–740. AAAI Press / The MIT Press, 2000.

[127] Janyce M. Wiebe, Theresa Wilson, Rebecca F. Bruce, Matthew Bell, and Melanie Martinr. Learning subjective language. *Computational Linguistics*, 30(3):277 – 308, 2004.

[128] Wikipedia. Folksonomy. http://en.wikipedia.org/wiki/folksonomy.

[129] Wikipedia. Genetic algorithm. http://en.wikipedia.org/wiki/genetic-algorithm.

[130] Yahoo! Directory. http://dir.yahoo.com.

[131] Hui Yang and Jamie Callan. Near-duplicate detection for erulemaking. dg.o 2005.

[132] Hui Yang and Jamie Callan. Near-duplicate detection for erulemaking. In *dg.o2005: Proceedings of the 2005 national conference on Digital government research*, pages 78–86. Digital Government Research Center, 2005.

[133] Jihoon Yang and Vasant Honavar. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems*, 13:44–49, 1998.

[134] Hong Yu and Vasileios Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In Michael Collins and Mark Steedman, editors, *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 129–136, 2003.

[135] Mohammed J. Zaki. Parallel and distributed association mining: A survey. *IEEE Concurrency*, 7(4):14–25, /1999.

[136] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. BIRCH: an efficient data clustering method for very large databases. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 103–114, Montreal, June 1996.

[137] Yongzheng Zhang, A. Nur Zincir-Heywood, and Evangelos E. Milios. Term-based clustering and summarization of web page collections. In *Canadian Conference on AI*, pages 60–74, 2004.

[138] Ying Zhao and George Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 515–524, New York, NY, USA, 2002. ACM Press.

# Glossary

## A

**AFS** Active feature selection, which utilizes the proposed rule and general-topic document collection as background knowledge to help perform feature selection.

**ASM** Adaptive similarity measure, in which the feature weight can be changed based on needs.

## C

**CD** Cluster description, which consists of a small set of carefully selected terms.

**CDD Measure** A surrogate measure for efficiently constructing informative CDs. The measure is defined in terms of the three factors of coverage, disjointness and diversity.

**CU Measure** A surrogate measure for efficiently constructing informative CDs. The measure is defined based on Category Utility function.

## E

**ERFR** E-rulemaking feedback repositories.

## O

**OIS-based Approach** An integrated approach that is based on three important factors of opinions (O), issues (I) and stakeholders (S) to construct SD for large ERFRs.

## P

**Pagoda Algorithm** A layer-based replacement algorithm to search for good CD terms.

## R

**RAPDC Approach** An effective approach to select RAs based on three important factors of popularity, diversity and cross-cluster coherence.

## S

**SCD** Succinct cluster description, which consists of a small set of carefully selected terms along with some other informative information.

**SD** Summaritive digest, which consists of three components: a clustering structure (either hierarchical or flat), succinct cluster descriptions (SCDs) and representative arguments (RAs) for each cluster.

This page was intentionally left blank