TOPIC DETECTION USING

MAXIMAL FREQUENT SEQUENCES

YAP YANG LENG, IVAN

(B.Eng, (Hons.), NUS)

A THESIS SUMBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2004

ACKNOWLEDGEMENTS

Undertaking this project has been a challenging but rewarding experience, and I wish to acknowledge the following persons who have made it all possible.

- My Lord Jesus Christ, whose strength is made perfect in my weakness.

- A/Prof Loh Han Tong, for his supervision, guidance, and encouragement in the course of this Masters programme, and for giving me the flexibility to work very independently.

- Shen Lixiang, for his help in the direction of the project, and for his suggestions in organizing and wording much of the material in this thesis.

- Jonathan Lim, for his help in deciphering many algorithms in the early stages of the project.

- Fellow research students Liu Ying, He Cong and Zhan Jiaming, for the information shared during the entire course of the project, and for the friendship built up over lunches, dinners and time spent in DTI.

- My family, for their patience, understanding and support through my entire Masters programme.

- And finally, my friends, Jeffrey Lau, Norman Lee, Koh Yi-leen, Hwang Jeong-Ki, Tan Su-lin, Karen Chen and many other friends in Grace Methodist Church, for their encouragement, chastisement during times of temporary slack, for discussing and proof-reading my work, and for their friendship.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

One key piece of useful information for a newly assembled document collection is the distinct topics contained within. Arriving at this list of topics is usually challenging, especially if it is to be done by manually reading through and comparing all the documents in the collection. This is thus one of the challenges facing users and administrators of document collections: finding the distinct topics within the datasets in an accurate and automated manner.

The purpose of this project is to investigate the usefulness of using Maximal Frequent Sequences (MFSs) as building blocks in identifying distinct topics in a dataset. In existing work, MFSs have been found to function as content descriptors of the documents in which they occurred in, and thus, we also investigate the usefulness of using the MFSs contained within topic clusters as topic descriptors.

The topic detection method we have implemented is a hybrid of an existing word sequence extraction algorithm – the MFS Discovery algorithm – and a heuristic to further group the MFSs into topic clusters. We carried out experiments on documents from two datasets, the Reuters-21578 news collection, and Manufacturing Corpus Version 1 (MCV1). Our experimentation involved the variation of two parameters associated with the algorithm: support and confidence. Firstly, we established suitable parameter values from our experiments on a subset of the Reuters-21578 dataset, and thereafter tested the suitability of these values on another subset of Reuters-21578. We then ran the algorithm with the same parameter values on several subsets of MCV1.

Our observations have led us to conclude that our method is best able to fulfill its purpose of topic detection when used on a dataset where the topics are well separated in terms of subject matter. For datasets whose documents have distinct topics but have an underlying association to begin with, the method is still able to work, but does not measure up to the performance in the former case.

For real world document datasets that do not have simply defined topic structures, our method is still useful. From our experiments on a subset of MCV1, we see that our method is able to generate a list of distinct topics that can act as an intermediate result to understanding and further partitioning the dataset.

CHAPTER 1 - INTRODUCTION

Given the exponential growth of the amount of electronic textual data available in the world, in the likes of news feed collections, archives of research papers, databases of customer survey responses etc, there is a need for data mining tools to help us extract valid, previously unknown, comprehensible and actionable information [1]. These tools can come in the form of automated or semi-automated techniques to make meaningful sense out of all the data. When dealing with a collection of documents, a vital piece of information that would add knowledge to the user would be the list of distinct topics present in the collection. Specifically, this would mean that we should find out the number of distinct topics present, as well as generate a topic descriptor for each topic.

The concept of clustering has existed even before the development of advanced data mining tools. In a child's formative years, he learns to group the objects he comes across as "toys", "food", "clothes" etc. by virtue of the fact that there are certain attributes for items in a category that identify them as being of that category, and dissimilar to objects in other categories. In the same manner, clustering as we know it today (in the context of data mining) was developed to group instances of data (in a dataset) that were similar together. The measure of similarity could be a "distance", where instances of data that are "close" together would form a cluster. Traditional methods of clustering include partitioning and hierarchical clustering methods. These methods will be covered in greater detail in Chapter 2.

Clustering has been used on numeric-attribute data with a certain amount of success, but before applying clustering methods on textual data to perform document

clustering, the difference between numeric-attribute data and textual data warrants some re-consideration in the way we apply our methods. For example, Euclidean distance, which is usually used as a measure of dissimilarity in clustering numeric-attribute data, is found to be unsuitable for text [2]. In Chapter 2, other than presenting traditional clustering methods, we also mention how they have been adapted to handle textual data.

Clustering has been used to perform topic detection in textual datasets with a fair measure of success [3, 4], using adaptations of various degrees of traditional clustering methods. In Seo and Sycara's work [3], the robustness of the constructive competitive clustering algorithm they developed is seen, when it showed an even trend of performance across data from different domains.

As seen from the previous work mentioned, clustering has been adapted for use on textual datasets, to perform document clustering, and also topic detection. This leads us to the motivation behind this project, to develop a method that is able to perform topic detection, through the use of clustering concepts. The purpose of this project is to investigate the usefulness of using Maximal Frequent Sequences (MFSs) as building blocks in identifying distinct topics of a textual dataset. The supporting documents of MFSs that have been grouped into a topic cluster function as a document cluster, that is representative of the documents in that topic. We also examine the usefulness of using MFSs as topic descriptors for each of the discovered topic clusters.

To fulfill the purpose outlined above, our proposed method to discover the distinct topics is a hybrid of an existing word sequence extraction algorithm, the Maximal Frequent Sequence (MFS) Discovery algorithm, and a heuristic to further group MFSs into topic clusters. In the later part of Chapter 2, we introduce the concept of MFSs and equivalence classes; Chapter 3 describes the implementation details of the original MFS Discovery algorithm, to extract MFSs from a document collection, and also the details of grouping MFSs into equivalence classes. Chapter 4 describes the hybrid method we adopted, to produce topic clusters.

The textual datasets that were used in this project include the publicly available Reuters-21578 news collection [5], and a collection of research papers assembled by Ying Liu et el. that deal with different aspects of Manufacturing [6]. The document collections differ in terms of content, with the latter being more relevant in the context of the engineering industry. The experimentation details and results for the Reuters dataset will be covered in Chapter 5, and the results for the Manufacturing research papers will be evaluated in Chapter 6 and 7.

The conclusions we arrive at for this project will be presented in Chapter 8, along with suggestions for possible future work.

CHAPTER 2 – LITERATURE REVIEW AND BACKGROUND STUDY

2.1 TOPIC DETECTION

In recent years, topic detection in text has been an actively researched subject, with more emphasis being given to detecting topics in a stream of data rather than a static dataset. In [7], a stream of broadcast news stories is being used, and the scope of the report encompasses the tracking of the detected topics as well. Chat lines in Internet Relay Chatrooms were used as data streams to be investigated in [8], to identify participants' topics of interest.

In our use of topic detection, we differ from the field in that our datasets are static; they include a collection of technical papers, which will not be similar in terms of structure and content to the dynamic data streams mentioned above. Hence, we seek to develop a method that is better suited for performing topic detection on the datasets we will be using.

2.2 CLUSTERING

In this section, we introduce how documents in a textual dataset are usually represented, before the application of clustering methods. We also cover two broad classes of clustering methods, partitioning and hierarchical methods, which have been applied to textual datasets to do document clustering.

2.2.1 Document Representation

Before the clustering methods can be applied to a dataset, the documents in a textual dataset have to be appropriately represented. As pointed out by Aas and Eikvil in [9],

the vector space model is the most commonly used document representation, where documents are represented by vectors of words. To represent a collection of documents, the different document vectors are lined up alongside to form a term-by-document matrix. Figure 2.1 shows a simple example of such a matrix, where a_{ik} is the weight of word *i* in document *k*.

$$\begin{array}{c} \\ \text{word } 1 \\ \text{word } 2 \\ \text{word } 3 \end{array} \begin{array}{c} \begin{array}{ccc} \text{doc 1} & \text{doc 2} & \text{doc 3} \end{array} \\ \begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & a_{32} & a_{33} \end{pmatrix} \end{array}$$

Figure 2.1. Example of a term-by-document matrix

In the matrix, document 1 is expressed as a column vector, containing only word 1, document 2 contains words 2 and 3, and so on.

A document weighting scheme has to be decided upon, to determine the values that the weights *a* will take in the matrix. Different approaches have been taken to determine *a*, but they are based on two empirical observations regarding text [9]:

- The more times a word occurs in a document, the more relevant it is to the topic of the document.
- The more times the word occurs throughout all documents in the collection, the more poorly it discriminates between documents.

A collection of weighting schemes of varying complexities is introduced in [9].

2.2.2 Partitioning Clustering Method

As the name suggests, partitioning clustering methods work by partitioning a set of objects into groups, called clusters, such that similar objects are in the same cluster, and dissimilar objects are in different clusters. These clusters are formed to optimize an objective partitioning criterion, often called a similarity (or dissimilarity, if the value increases with dissimilarity between clusters) function, such as distance [10].

One of the most commonly used partitioning methods is the *k*-means method. The method takes in an input integer parameter *k*, and partitions a set of *n* objects into *k* clusters so that the resulting intracluster similarity is high but the intercluster similarity is low. Objects are represented by points in metric space, and a cluster is represented by an aggregate of the objects within. As mentioned earlier, the dissimilarity function is usually a distance, and typically, the Euclidean distance between the objects to be compared is used.

In using *k*-means to cluster documents in a textual dataset, a term-by-document matrix, which was described in Section 2.2.1, is used to represent the dataset. *k* document vectors are initially chosen, and the remaining document vectors are assigned to these *k* documents, based on how similar they are to each other. In its traditional use, the *k*-means method uses Euclidean distance as a measure of dissimilarity between data points, but this distance measure is found to be inappropriate for document clustering [2]. A better measure of document similarity is cosine similarity, which is calculated by taking the cosine of the angle between the two document vectors to be compared [11]. Once every document has been assigned to one of the *k* document clusters, the cluster means will be evaluated, and the

documents are once again re-assigned, to the cluster mean they are most similar to. This re-assignment and re-calculation of cluster means continues until there is no more change in document assignment.

A variation of the *k*-means method is the *k*-medoids method. Instead of using the cluster mean as a reference point, the medoid, which is the most centrally located object in a cluster, is used. The medoid acts as a representative of the cluster contents, which offers some form of description of the cluster.

In its use in textual clustering, *k* documents are again initially chosen, and the remaining documents are assigned to these *k* documents, based on the cosine similarity between the objects. These *k* objects would be the initial medoids. We next randomly select a non-medoid object, and compute the difference in cost of swapping a medoid object o_i of cluster C_i with the non-medoid one. The cost function is given as:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} \left| p - o_i \right|^2 \qquad (2.1)$$

where E is the sum of squared-error, p is the point in space representing a given object, and o_i is the medoid of cluster C_i (where p and o_i are both multi-dimensional). If the difference in cost is negative, the swap is made, to form a new set of *k* medoids. The documents in the dataset are once again re-assigned to the nearest medoids, and another randomly chosen non-medoid object will be subjected to the same procedure. The process of re-assignment of medoids continues until there is no more change.

For a method like *k*-means, the value of the input parameter *k* is assumed to be known [12, 13], and the final clustering solution is dependent on the order in which the documents were processed, and the random selection of documents as initial cluster centers. The dependence of the solution on these factors leads to a rather arbitrary classification of documents into their respective clusters [14]. Moreover, after the clusters are generated, there are no descriptors available to define or describe the content within each cluster.

The *k*-medoids method is more robust with respect to outliers, compared to the *k*-means method, but being a variation of the *k*-means method, it also suffers from similar drawbacks. Both methods favour clusters that are spherical-shaped, and the input parameter *k* is assumed to be known. A possible way to deal with the latter problem is found in Kaufman and Rousseeuw's implementation of a *k*-medoid algorithm, PAM (Partitioning Around Medoids) [12]. The user is allowed to specify a range of values for *k* at the onset of the algorithm, and thereafter evaluate which *k* is most suitable. In the described adaptation of *k*-medoids for text clustering, the medoid is a unique and distinct document, and while it may be representative of the cluster, we still do not have a breakdown of the word phrases that actually give the cluster its distinct identity.

2.2.3 Hierarchical Clustering Method

Hierarchical clustering methods work by grouping data objects into a tree of clusters. The methods are further classified into agglomerative and divisive hierarchical clustering, depending on whether the hierarchical decomposition is formed in a bottom-up or top-down fashion [10].

In hierarchical clustering, we present three ways to calculate the intercluster dissimilarity: single linkage, complete linkage and group average linkage. The Euclidean distance is first calculated, between pairs of members in the respective clusters. For single linkage, the intercluster distance is defined as the minimum of these distances; for complete linkage, the intercluster distance is the maximum of the same set of calculated distances. For group average linkage, the intercluster distance is taken to be the average of all the distances between all pairs of members in the clusters that are being considered. These three linkage methods are useful in different types of applications; the group average method favours ball-shaped clusters [12], the single linkage method can delineate nonellipsoidal clusters [15], and the complete linkage method is able to identify compact clusters that not well separated [12].

In applying agglomerative hierarchical clustering to a textual dataset, cosine similarity is usually used as a measure of similarity [16, 17], and each document in the dataset is initially considered as a cluster. For a textual dataset with N documents, an N x N interdocument similarity matrix is generated, to compare the closeness of a document with every other document in the dataset. Cosine similarity is initially computed between the documents, and the similarity matrix is then updated with the values. A similar approach is taken in part of the methodology in [18]. The two documents that are most similar are merged to form a new cluster, and intercluster similarity is re-evaluated, between the resulting set of (N-1) clusters, using an appropriate linkage method. The process of merging and re-evaluation of intercluster similarity continues until some stopping criterion is achieved. In divisive hierarchical clustering, the process is reversed; all the documents are initially contained in one big cluster, and

subsequently subdivided into smaller clusters based on dissimilarity between clusters, until some stopping criterion is achieved. For both variations of the method, the stopping criterion could be a target number of clusters found, or the intercluster distance between the two closest clusters not exceeding a certain threshold.

In certain situations, the stopping criterion is not easily defined; the domain knowledge might not be sufficient to determine how many clusters the algorithm should terminate at. There is also no descriptor for the content in each cluster, after the algorithm terminates.

2.2.4 Inadequacy of term-by-document matrix representation

Other than the individual drawbacks of each method as pointed out in Sections 2.2.2 and 2.2.3, there exists a common disadvantage. Both the partitioning and hierarchical clustering methods represent the textual dataset in a term-by-document matrix. Although the significance of the words in the documents may be conveyed by the use of a weighted document representation scheme, like the *tf-idf* (term frequency-inverse document frequency) weighting scheme [9], the document matrix is unable to represent the order of the words in the documents. If there is any information to be extracted from the order in which the words appear in the documents, these two methods are unable to do it.

2.3 MAXIMAL FREQUENT SEQUENCES (MFSs) AND EQUIVALENCE CLASSES

2.3.1 Maximal Frequent Sequences (MFSs)

The idea and method of discovering the set of MFSs out of a textual dataset was first proposed by Ahonen [19]. A MFS is a sequence of words that is "frequent in the document collection and, moreover, that is not contained in any other longer frequent sequence" [19]. A word sequence is frequent if it appears in at least σ documents, where σ is a pre-specified support threshold. The goal of the MFS algorithm is to find all maximal frequent phrases in the textual dataset.

The strength of the method is that it employs a versatile technique for finding sequential text phrases from full text, allowing, if desired, gaps between the words in a phrase [20]. For example, the word sequence "*product knowledge databases*" can be extracted as a frequent phrase even if its occurrence is in the form of:

- "*…product management using knowledge databases…*"
- "*…product data in knowledge databases…*"
- "*…product specifications, knowledge databases…*"

in the supporting documents of the document collection. The maximum gap allowed between words of a sequence is determined by the *maximal word gap* parameter.

In its original application, the MFSs discovered acted as content descriptors of the documents in which they occurred [19, 20]. These descriptors are compact but human-readable, and have the potential to be used in subsequent analysis of the documents.

2.3.2 Equivalence Classes

The definition for an equivalence class is described as such: “Phrases X and Y belong to the same equivalence class if they are descriptive of almost the same documents.” [20]. Under these circumstances, we can also say the equivalence class of X is equal to the equivalence class of Y. An additional parameter, *confidence*, is required. In Ahonen’s implementation, confidence was set at 0.9. The details of how the confidence parameter is used will be covered in greater detail in Section 3.3 in Chapter 3.

The original purpose of equivalence classes, as implemented by Ahonen [20], was to group MFSs that were similar together as a single entity, to reduce the number of distinct MFSs. This is desirable if the set of MFSs is be used as input into subsequent applications, e.g. frequent set generation among MFSs. A reduction in the number of MFSs (itemsets) would benefit the computational process of working out the solution.

CHAPTER 3 – THE MFS DISCOVERY ALGORITHM WITH EQUIVALENCE CLASSES

3.1 EXPLANATION OF MAXIMAL FREQUENT SEQUENCE (MFS) DISCOVERY

In this section, we flesh out the concept of MFS Discovery, and also explain what each of the terms mean, and how they are applied. Examples will be used to illustrate the basic concept; the detailed implementation of the Discovery algorithm will be covered in the Section 3.2.

Some concepts within the application of the MFS algorithm are introduced as follows:

Given a set of documents *S*, each document containing a sequence of words:

Definition 3.1. A sequence $p = a_1 \ldots a_k$ is a *subsequence* of a sequence *q* if all the items a_i, $1 \leq i \leq k$, occur in *q* and they occur in the same order as in *p*. If a sequence *p* is a subsequence of a sequence *q*, we also say that *p occurs* in *q*.

Definition 3.2. A sequence *p* is frequent in *S* if *p* is a subsequence of at least σ documents of *S*, where σ is a given *support threshold*.

We only count one occurrence of a sequence in a document; several occurrences within a document do not make the sequence more frequent. However, such multiple occurrences need to be recorded, to facilitate possible subsequent expansion.

Definition 3.3. A sequence *p* is a *maximal frequent (sub)sequence* in *S* if there does not exist any sequence *p'* in *S* such that *p* is a subsequence of *p'* and *p'* is frequent in *S*.

Table 3.1 shows a sample document collection, where each row is considered a document. The text has been lower-cased, and the punctuation removed, so that word sequences that are alike across documents can be identified.

Table 3.1. Example of a Textual Dataset (lower-cased, punctuation removed)

ID	Document Contents
1	the two most well known models from fender guitars are the strat and the tele
2	gibson is famous for their les paul and sg guitars
3	many professional guitarists have among their guitars a strat tele les paul and sg
4	brands well known for their acoustic guitars include taylor and martin
5	classical guitars differ from acoustics and electrics in that they use nylon strings

When we set the support threshold $\sigma = 2$, it means that the word sequences must appear in at least two documents for them to be frequent. Looking at documents 2 and 3 in Table 3.1, we see that when support = 2, the phrase "*les paul*" is frequent, because it appears in these two documents.

Take note that "*les paul*" is not a MFS, because if we examine the dataset, it is part of the longer frequent phrase "*les paul and sg*", which also appears in documents 2 and 3. This observation is highlighted in Table 3.2, where the frequent phrase is in bold, and the MFS is highlighted. This longer phrase "*les paul and sg*" is a MFS, firstly because it is frequent (Definition 3.2), and secondly, because it is not contained in any other longer frequent sequence (Definition 3.3). This can be verified by examining the dataset manually.

Table 3.2. Example of a frequent phrase and a MFS in the dataset

ID	Document Contents
2	gibson is famous for their **les paul** and sg guitars
3	Many professional guitarists have among their guitars a strat tele **les paul** and sg

To make the MFS extraction more efficient, we restrict the maximal distance of two consecutive items in a word sequence. Looking at document 1 in Table 3.1, it would not be meaningful to consider the first and last words of the document, "*the*" and "*tele*", as a word sequence, because intuitively, we know that words that are far apart in a sentence or a paragraph do not carry much meaning when they are grouped together.

Definition 3.4. The *maximal word gap g* is used to restrict the distance between consecutive words of a word sequence. This means that at most *g* other words may be between the consecutive words of a word sequence.

Based on the literature [19, 20], maximal word gap is set as 2. To illustrate its use, assume we set support as 2, and maximal word gap *g* to be 2. We now examine the phrase "*strat tele*", which appears in documents 1 and 3 in the dataset. Table 3.3 contains these two documents, for closer analysis. In document 1, even though the two words do not appear consecutively, the number of words in between does not exceed the maximal word gap of 2, and thus these two words can be grouped as a potentially frequent phrase. It is verified to be frequent when we see that "*strat tele*" also appears in document 3. In the same manner, we can identify "*guitars strat tele*" as a MFS, which is highlighted in Table 3.3.

Table 3.3. Example of a frequent phrase and a MFS in the dataset; maximal word gap = 2

ID	Document Contents
1	the two most well known models from fender guitars are the **strat** and the **tele**
3	many professional guitarists have among their guitars a **strat tele** les paul and sg

Thus, the MFSs that are associated with the document collection given in Table 3.1, with a support of 2 and a maximal word gap of 2, are:

- "*guitars strat tele*"
- "*les paul and sg*"
- "*well known*"

These results can be verified by examining the dataset in Table 3.1 manually.

3.2 BREAKDOWN OF STEPS FOR MFS DISCOVERY

In this section, we break down the steps that need to be implemented, to discover the set of MFSs of a textual dataset. Algorithms 1 and 2 describe the MFS discovery algorithm in pseudo-code, and will be elaborated upon in this section.

Algorithm 1 takes in a set of pre-processed documents *S*, a support threshold σ, and a maximal word gap *g* as input, and returns a set of MFSs. There are two main phases to the algorithm: the *Initial* phase and the *Discovery* phase.

- Initial Phase

This phase collects all the frequent pairs from *S* (Alg. 1 line 1-3). We start by collecting all the ordered pairs, or 2-grams, within *S* (Alg. 1 line 2-3). The maximal

Algorithm 1: Discovery of all maximal frequent sequences in the textual dataset

Input: S : a set of pre-processed documents, σ : a support threshold,
g : maximal word gap
Output: *Max* : a set of maximal frequent sequences

// Initial phase: collect all frequent pairs
1. For all the documents $d \in S$
2. collect all the ordered pairs and occurrence information within d
3. $Grams_2$ = all the ordered pairs that are frequent in S
// Discovery phase:
// build longer word sequences by expanding and joining,
// and store MFSs and occurrence information into *Max*
4. $k := 2$
5. $Max := \theta$
6. While $Grams_k$ not empty
7. For all grams $g \in Grams_k$
8. If g is frequent
9. If g is not a subsequence of some $m \in Max$
// Expand phase: expand frequent gram
10. max := Expand(g) // Refer to Algorithm 2
11. $Max := Max \cup max$
12. If $max = g$
13. Remove g from $Grams_k$
14. Else
15. Remove g from $Grams_k$
16.
// Join phase: generate set of (k + 1)-grams
17. $Grams_{k+1}$:= Join($Grams_k$)
18. $k := k + 1$

Figure 3.1. Algorithm 1, MFS discovery algorithm

Algorithm 2: Expand

Input: p : a sequence
Output: p' : a maximal frequent sequence such that p is a subsequence of p'

1. Repeat
2. Let l be the length of the sequence p.
3.
4. Find a sequence q' such that the length of q' is $l + 1$,
5. and p is a subsequence of q'.
6.
7. If q' is frequent
8. $p := q'$
9.
10. Until there exists no frequent q'
11. $p' = p$

Figure 3.2. Algorithm 2, MFS discovery algorithm

word gap parameter *g* is used to limit the number of pairs that each document can produce. A single pass across all the documents in the dataset is required, to collect the ordered pairs for each document. Each ordered pair is stored into a hash data structure, along with its occurrence information. The occurrence information consists of the document IDs in which the phrase occurs, and the word IDs that define the position of the phrase in the document. Figure 3.1 contains an example of the occurrence information for the phrase "*les paul*".

ID	Document Contents
2	gibson is famous for their **les paul** and sg guitars
3	many professional guitarists have among their guitars a strat tele **les paul** and sg

Occurrence information of "*les paul*"

- 2:6-7 (occurs in doc_id 2, occupying positions in word_id 6 and 7)
- 3:11-12 (occurs in doc_id 3, occupying positions in word_id 11 and 12)

Figure 3.3. Occurrence information of the frequent phrase "*les paul*"

The collection of the ordered pairs is repeated for all the documents in *S*, with a corresponding update of the hash structure which is storing the occurrence information. Thereafter, each ordered pair in the hash is examined, and the pairs that are supported by at least σ documents are considered frequent. This set of frequent pairs is named $Grams_2$ (line 3).

- Discovery Phase

The next phase, the *Discovery* phase (Alg. 1 line 4-18), forms the main body of Algorithm 1. It is an iteration of gram expansion for the grams in the current $Grams_k$,

and gram joining, to form $Grams_{k+1}$. Only grams that are frequent and not subsequences of some previously discovered MFS (Alg. 1 line 8-9) are considered suitable for expansion. The latter condition is in place to avoid a rediscovery of MFSs that have already been found. This Expand-Join iteration continues until an empty gram-set is produced from after a Join phase.

We further break the *Discovery* phase into the *Expand* phase (Alg. 1 line 10-16, Alg. 2) and the *Join* phase (Alg. 1 line 17-18).

In the *Expand* phase, grams form $Grams_k$ are input iteratively into Algorithm 2 (Alg. 1 line 10), and every possibility of expansion of an input sequence p is explored [1]. In the first iteration, when $k = 2$, we take each frequent pair from $Grams_2$ and expand it in a greedy manner until the phrase is no longer frequent.

Algorithm 2 outlines the expansion process of a gram p of length l. The goal of the exploration of expansion possibilities is to find a frequent gram, that is one word longer than the previous frequent gram that was used as input. When a frequent gram p is first input to Algorithm 2, we try adding a new word to the tail first, and if the resulting (l+1)-gram is not frequent, we will try to add that same word to the head, and if the result is still a non-frequent (l+1)-gram, we will try to add to the middle of the gram. When a frequent (l+1)-gram is discovered in any of the three scenarios, the algorithm proceeds by trying to further expand the new frequent gram, and does not consider what other frequent (l+1)-grams can arise out of the original gram p (Alg. 2 line 1, 7-8). This greedy manner of expansion continues, for that particular input sequence p, until the gram is no longer frequent (Alg. 2 line 10).

The last frequent sequence achieved in the expansion, *p'*, will be an MFS by definition, and it will be returned to the calling function, together with its occurrence information (Alg. 1 line 10). These MFSs, along with their occurrence information, will be stored in another hash *Max* (Alg. 1 line 11). This process of gram expansion and information recording continues, for every suitable gram in $Grams_k$. When expanding the grams in $Grams_k$, k-grams that cannot be further expanded are themselves MFSs, and they will be removed from the set.

Throughout the *Expand* phase, the gram to be expanded and the MFSs already discovered are tracked by their document and word IDs as illustrated in Figure 3.1, so that we are able to specifically reference a word or gram in a document, even if it occurs multiple times in one document. This system of referencing allows us to limit the number of words to be checked in the vicinity of a gram in each document, in verifying whether a longer gram is possibly frequent.

The *Join* phase follows, which consists of a simple join operation amongst the grams left in $Grams_k$, to form $Grams_{k+1}$, i.e. the set of grams that are of length $(k+1)$. The value for *k* is correspondingly increased by 1, and if the new gram set is not empty, the grams will be expanded in the same manner. When an empty gram set is produced from the Join phase, Algorithm 1 proceeds to line 19, and the set of MFSs is returned.

3.3 EQUIVALENCE CLASSES IN MFSs

In this section, we describe the method for grouping MFSs together into an equivalence class.

Definition 3.5. Let *A* and *B* be two MFSs amongst the set of MFSs discovered. The *equivalence class* of *A*, Eq_A, contains the set of MFSs that co-occur with *A* in almost the same documents, as given by a *confidence* parameter. Det_A is the set of MFSs that are *determined by A*, and is required in deciding which MFSs belong in Eq_A.
For MFSs *A* and *B*, if:

$$\frac{Frequency(A, B\ co-occur)}{Frequency(A)} \geq Confidence \qquad (3.1)$$

then we add *B* to the set Det_A; *A* itself is also included in Det_A. The other MFSs are tested in the same manner, and will be added to Det_A if they satisfy the above criterion. Eq_A is thus made up of all MFS *X* such that $Det_X = Det_A$.

To generate equivalence classes, we require as input:

- the set of MFSs that was returned from the MFS Discovery algorithm, and,
- a confidence parameter.

Table 3.4 contains a hypothetical set of four MFSs that have been returned from the MFS Discovery algorithm, and the supporting documents for each MFS. Before the equivalence classes can be obtained, we need to find the *Det* sets for each of the MFSs.

Table 3.4. Sample set of MFSs with supporting documents

MFS	**IDs of Supporting Documents**
A	1 2 3 4 5 6
B	2 3 4 5 6 7
C	3 4 5 6 7 8
D	2 3 4 5 9 10 12 13

Assuming confidence = $\frac{4}{6}$, we first find which MFSs are determined by *A*.

Using the criterion specified in Definition 3.5,

	$\underline{Co-occurring\ Documents}$
$\frac{Frequency(A, A\ co-occur)}{Frequency(A)} = \frac{6}{6} \geq Confidence$	1,2,3,4,5,6
$\frac{Frequency(A, B\ co-occur)}{Frequency(A)} = \frac{5}{6} \geq Confidence$	2,3,4,5,6
$\frac{Frequency(A, C\ co-occur)}{Frequency(A)} = \frac{4}{6} \geq Confidence$	3,4,5,6
$\frac{Frequency(A, D\ co-occur)}{Frequency(A)} = \frac{4}{6} \geq Confidence$	2,3,4,5

Since all four MFSs meet the criterion, all four MFSs are determined by *A*, and are added to the set of Det_A. Therefore, $Det_A = \{A, B, C, D\}$.

The *Det* sets for the other MFSs are also evaluated in the same manner, and the results are shown in Table 3.5.

Table 3.5. Sample set of MFSs with supporting documents and *Det* sets

MFS	**IDs of Supporting Documents**	$\boldsymbol{Det_{MFS}}$
A	1 2 3 4 5 6	$Det_A = \{A, B, C, D\}$
B	2 3 4 5 6 7	$Det_B = \{A, B, C, D\}$
C	3 4 5 6 7 8	$Det_C = \{A, B, C\}$
D	2 3 4 5 9 10 12 13	$Det_D = \{D\}$

Evaluating the results according to Definition 3.5, we see that only for *A* and *B*, the *Det* sets match, i.e. $Det_A = Det_B$. Therefore, when we use a confidence value of $\frac{4}{6}$, only one equivalence class is generated from this set of MFSs, and this equivalence class is made up of MFSs *A* and *B*.

In previous work on association rule mining [21, 22], the confidence parameter is a measure of the strength of the rule. The use of confidence here is similar; for each MFS, we find other MFSs that are associated to it (with a strength as determined by confidence), and include them in its *Det* set. When we next group MFSs that have matching *Det* sets together, we are consolidating MFSs that are associated to the same set MFSs, and this consolidated group of MFSs we term an equivalence class.

CHAPTER 4 – TOPIC DETECTION USING MFSs AND EQUIVALENCE CLASSES

In this chapter, we describe our method of using MFSs and equivalence classes to detect the distinct topics in a document collection. The method is an adaptation of the original MFS Discovery and equivalence class extraction algorithms, which were described in Sections 3.2 and 3.3 respectively.

After a document collection is processed in the manner described in Sections 3.2 and 3.3, a set of equivalence classes is generated. Our method uses these equivalence classes as objects representative of the distinct topics in the dataset; Definition 4.1 provides the criterion for whether an equivalence class is representative of a topic.

Definition 4.1. An equivalence class is considered to be *representative* of a topic, if the *precision* of the class is at least ϕ, i.e. amongst the documents in an equivalence class, the proportion of supporting documents for a particular topic is of the equivalence class at least ϕ.

In our work, we set $\phi = 0.7$; at least 0.7 of the documents belonging to an equivalence class must be from one topic, before we deem that class to be representative of that topic.

A topic may be represented in more than one equivalence class, and so, we need to group classes that belong to the same topic together, into *topic clusters*. Thus, our method returns the following information regarding the document collection:

- the number of distinct topics in the document collection, which is simply the number of topic clusters detected;
- topic descriptors for each distinct topic, which are made up of the MFSs in the grouped equivalence classes; and,
- the occurrence information of all the MFSs found in the equivalence classes, from which we can obtain sample, representative documents of each topic cluster.

In the literature [19, 20], only one level of support was used in the MFS Discovery algorithm, and thereafter, only one level of confidence was used, to group MFSs together into equivalence classes. We depart from the literature in our method; we use a range of values for support and confidence instead of a single value. This is done to maximize the coverage of distinct topics amongst the output equivalence classes.

Given a document collection, our method follows the following step-by-step procedure. The procedure is illustrated in Figure 4.1:

– *Step 1: Pre-processing*

At this stage, we convert to lower-case all the characters in the text and remove punctuation symbols and numbers. Stopwords – common words (e.g. is, are, the) that appear across multiple documents and have little use in characterizing a topic – are also removed from the text in this stage. We use a stopword list of 429 words, that has been used in the development and implementation of a text indexing engine, Onix [23].

– *Step 2: Set parameters*

There are several parameters in the MFS Discovery algorithm that need to be specified. They are *support, maximal word gap* and *confidence,* and suitable values for each parameter will be chosen and implemented. We depart from the literature [19, 20] in our use of support and confidence parameters; instead of a single value for each, we now choose a range of values. Thus, each level of support would yield its own set of MFSs. Likewise, each value of confidence would yield its own set of equivalence classes, from the resulting set of MFSs from the earlier step. The use of a range of values, instead of single values, is done to maximize the topic coverage.

– *Step 3: Discover the set of MFSs of a textual dataset*

From the previous step, we have a range of support values. We begin MFS Discovery on the pre-processed dataset, beginning with a *single*, *starting value* of support. MFS Discovery is performed as described in Section 3.2, until the set of MFSs associated with the current level of support is generated. This set of MFSs is passed to the next step.

– *Step 4: Find the equivalence classes amongst the MFSs*

Using the range of confidence values we obtained in Step 2, we iterate the equivalence class extraction process over this range of values, based on the steps given in Section 3.3. Thus, we will use the same set of MFSs from the previous step as input, and extract a separate set of equivalence classes, for each value of confidence we have. These equivalence classes are stored in preparation for their

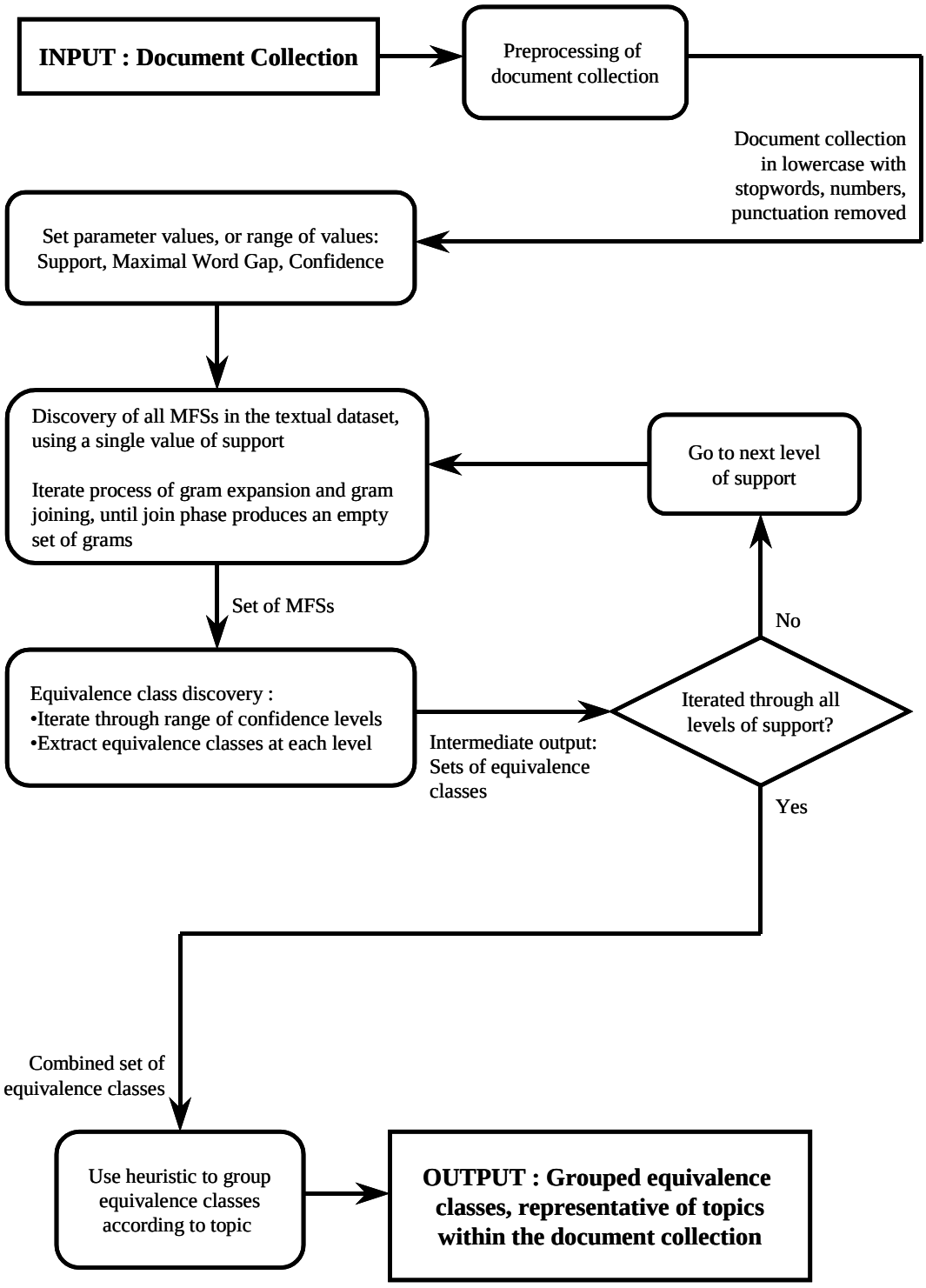

Figure 4.1. Flowchart illustrating implementation of our method to discover topics in a document collection

use in Step 5. After iterating through the confidence values, our method returns to Step 3, and repeats Step 3 and 4 for the next value of support, until the full range of support values have been iterated through. The combined set of equivalence classes from all the iterations is then passed on to the next step.

– *Step 5: Group the equivalence classes*

From the combined set of equivalence classes returned from Step 4, we group equivalence classes that are representative of the same topic together, into *topic clusters*. This forming of topic clusters is done by a simple heuristic of identifying equivalence classes that have co-occurring words, and grouping them together. Table 4.1 contains an example of equivalence classes that have been grouped together.

Table 4.1. Example of grouped equivalence classes

Equivalence Class 1	**Equivalence Class 2**
alcan ltd, aluminium ltd	aluminium company, finance minister

Looking at Table 4.1, we see that each equivalence class contains 2 MFSs, and all the MFSs are different. However, the word "*aluminium*" appears in each equivalence class, and hence it is a co-occurring word. Using our heuristic, we can group these 2 equivalence classes together, to form a topic cluster. A topic cluster like this contains descriptive phrases (the MFSs) of the topic content, and also the occurrence information of the maximal phrases, which can be used to identify the documents that belong to the cluster. However, the documents identified in each cluster are not the complete set of documents associated with the topic; they

function instead as a representative sample of what the documents belonging to the topic are like.

The end products achieved in our method are the topic clusters. The number of clusters gives us the number of distinct topics in the document collection, the supporting documents in each group provide sample, representative documents of the topic, and the MFSs in each group act as topic descriptors.

CHAPTER 5 – EXPERIMENTATION DETAILS AND RESULTS FOR REUTERS-21578 DATASET

In this chapter, we perform experiments on subsets of the Reuters-21578 news collection [5]. The purpose of these experiments is to show that our method is able to detect the distinct topics in a textual dataset, within a suitable, narrowed-down range of support and confidence parameters. As described in our hybrid method in Chapter 4, we need to establish suitable values for these parameters in our implementation of the algorithm. These values were chosen based on experiments performed on *Reuters_261*, a subset of Reuters-21578, and thereafter, the suitability of these parameter values was tested on *Reuters_299_mixed*, another Reuters-21578 subset. The maximal word gap was taken to be 2 in both sets of experiments.

5.1 DESCRIPTION OF DATASETS

The Reuters-21578 dataset is a set of 21,578 documents which appeared on the Reuters newswire in 1987. The documents were assembled and indexed with categories by personnel from Reuters Ltd. and Carnegie, Ltd. in 1987, and subsequently collected and formatted by David Lewis and his associates [5]. Each document is tagged with a series of meta-tags that store the document's index number, date, topic classification, place involved etc., amongst other things. We are primarily interested in the topic classification of each document, and the main body of text, which makes up the document.

The topic classification is done according to a list of 135 topics, mostly dealing with business and the economy. The full list of topics is found in Appendix A. Documents that have content that is associated with only one of the 135 topics are tagged with a single topic, whereas documents which cover a greater scope in their stories would

have multiple topic assignments. In the collection, there are also documents which have topic assignments, but no actual body of text; these documents we deem to be erroneous.

We extract documents from the Reuters-21578 dataset for our experimentation; we will be using documents that have both single and multiple topic assignments, but we will leave out the erroneous ones.

The goal for the first set of experiments was to choose a suitable range of parameter values for our method, which would yield a concise set of MFSs that would still cover all the distinct topics in the dataset. In this first dataset we assembled for this purpose, we chose only documents that had single topic tags, and had an actual body of text contained within. The dataset that we used for this stage of our experimentation, *Reuters_261*, consisted of 261 documents across a range of six topics[1]. The number of documents in each topic was between 41 and 46.

The second dataset, *Reuters_299_mixed*, was used to test the validity of the parameter values found using the first dataset. It consisted of 299 documents from five topics[2]. In this dataset, we included some documents that were tagged with more than one of the five topics, and tested if the method was still able to detect the distinct topics within the dataset. The number of documents that had single topics was 224, and the number of documents tagged with two topics was 75.

[1] alum, grain, ipi, iron-steel, nat-gas, reserves
[2] alum, crude, grain, nat-gas, ship

Further details with regards to the assembly of these two datasets can be found in Appendix B.

5.2 TOPIC DETECTION WITH SINGLE-TOPIC DOCUMENTS – *Reuters_261*

5.2.1 Experimental Procedure

Our criterion for appropriate values of support and confidence was as follows: among the equivalence classes generated from the target range of parameter values, all six topics of the dataset had to be *represented*, where 'represented' takes the meaning as specified in Definition 4.1.

To determine the appropriate levels of support to be used, we ran the MFS algorithm on the *Reuters_261* dataset across high to low support levels. Higher support levels produced MFSs that were shorter; we started with a support of 25, which only produced MFSs that were two words long. From this starting value, we lowered support in steps of 1, to a support level of 5, which we set to be the lower limit for how frequent a phrase had to be, to be representative of a topic.

For each level of support, we varied the level of confidence from 0.9 to 0.1, in steps of 0.1, and we examined the equivalence classes found. We looked at the equivalence classes for various levels of confidence, and chose combinations of support and confidence that had a wide spread of topics represented in the equivalence classes.

5.2.2 Experimental Results

The results from the *Reuters_261* dataset are summarized in Table 5.1, which shows the number of topics represented for each combination of support and confidence.

Only the results for support 9 to 5 are shown, where all 6 topics are represented. The results for the other levels of support are not shown because they did not have comprehensive topic representation. These combined results take into account the different confidence levels at each level of support.

Table 5.1. Topic spread of equivalence classes from support 9 to 5, confidence 0.9 to 0.1, for *Reuters_261*

Confidence	Number of topics present				
	support 9	support 8	support 7	support 6	support 5
0.9	0	1	0	0	2
0.8	4	2	4	3	5
0.7	4	4	3	3	6
0.6	4	2	5	4	4
0.5	5	3	4	4	4
0.4	3	3	3	3	2
0.3	1	3	2	2	2
0.2	1	3	2	2	2
0.1	0	2	1	1	3
Topics found in confidence 0.8 to 0.5	1,3,4,5,6	2,3,4,5,6	2,3,4,5,6	2,3,4,5,6	1,2,3,4,5,6

Looking at Table 5.1, for support levels between 9 to 5, the range of confidence values from 0.8 to 0.5 gave a complete spread of the topics in the dataset. The results for this range of parameters are boxed-up in Table 5.1, and this range is chosen because topic coverage is maximized amongst the equivalence classes.

Some equivalence classes were replicated across confidences and supports, so we took the union of the equivalence classes found from support 9 to 5, confidence 0.8 to 0.5, to form a combined set of distinct equivalence classes.

It was observed that equivalence classes that belonged to the same topic had certain words or phrases that overlapped across classes. A simple heuristic of grouping

equivalence classes with co-occurring words/phrases was used to group the classes into topic clusters, to get the distinct topics in the dataset, as well as topic descriptors for each topic. Table 5.2 provides an example of how the equivalence classes can be grouped together to identify the distinct topics, where the highlighted words are the co-occurring terms, which act as links between the equivalence classes in each topic.

Table 5.2. Sample equivalence classes from each topic in *Reuters_261*

Equivalence Class 1	**Equivalence Class 2**	**Actual Topic**
alcan ltd, aluminium ltd	aluminium company, finance minister	alum
grain harvest, grain mln tonnes	mln grain, mln tonnes grain	grain
industrial index base, industrial production base, industrial production index	figure ministry, figure revised, industrial production rose pct revised pct	ipi
bethlehem steel corp, dlrs ton	lt steel, steel corp lt	iron-steel
cubic feet natural gas, cubic natural gas	energy co, inc gas, gas corp, gas pipeline, gas sales	nat-gas
currency billion, currency reserves	gold mln, gold reserves mln	reserves

However, the grouping based on this heuristic alone is imperfect, as certain equivalence classes may fulfill this criterion but do not belong to the same topic. For example, the word '*mln*' (meaning *million*) appears in topics "*grain*" and "*reserves*", but the word is not the main subject of the equivalence classes it appears in. In future, we are looking to improve the heuristic to perform this grouping of equivalence classes automatically.

Looking at the MFSs within each group of equivalence classes in Table 5.2, we also see that the MFSs are related to the actual topics they are supposed to represent. For

example, the topic "*ipi*" stands for "industrial production index", an economic indicator code. Looking at the MFSs within this particular group, we see that the phrases are related to industrial production, and also suggest the involvement of an index of sorts ("*figure revised, rose pct revised pct*").

For some combinations of support and confidence, there were instances where bad equivalence classes were generated, i.e. the supporting documents did not distinctly come from any one topic, and as such, were not representative of any topic. However, the error rate was low across most combinations. Table 5.3 shows that for our chosen ranges of support and confidence, the number of bad equivalence classes generated at each parameter combination is generally a small percentage of the good equivalence classes, and so, the grouping of the good classes into clusters was relatively unhindered by the presence of the bad classes.

Table 5.3. Percentage of good equivalence classes, from support 9 to 5, confidence 0.8 to 0.5, for *Reuters_261*

Support 9					Support 6				
confidence	No. of good classes	No. of bad classes	Total no. of classes	% good	confidence	No. of good classes	No. of bad classes	Total no. of classes	% good
0.8	6	1	7	85.7	0.8	8	0	8	100
0.7	7	1	8	87.5	0.7	19	0	19	100
0.6	7	1	8	87.5	0.6	16	0	16	100
0.5	6	1	7	85.7	0.5	6	1	7	85.7
Support 8					Support 5				
confidence	No. of good classes	No. of bad classes	Total no. of classes	% good	confidence	No. of good classes	No. of bad classes	Total no. of classes	% good
0.8	2	1	3	66.7	0.8	20	1	21	95.2
0.7	7	2	9	77.8	0.7	26	2	28	92.9
0.6	3	2	5	60.0	0.6	15	2	17	88.2
0.5	5	2	7	71.4	0.5	16	3	19	84.2
Support 7									
confidence	No. of good classes	No. of bad classes	Total no. of classes	% good					
0.8	5	0	5	100					
0.7	7	0	7	100					
0.6	13	0	13	100					
0.5	7	0	7	100					

From the results, we have determined the suitable values to be used for the parameters in our method; support values from 9 to 5, and confidence values from 0.8 to 0.5. These values are a guideline for us to further investigate the usefulness of using grouped equivalence classes of MFSs as topic clusters.

5.3 TOPIC DETECTION WITH MULTI-TOPIC DOCUMENTS – *Reuters_299_mixed*

It is seen that our method is effective in distinguishing single-topic documents, and it able to do so with a narrowed-down range of parameters. In this section, we will apply the method to multi-topic documents, since real world document collections usually contain documents that span across a few topics.

We tested the suitability of the parameter values by repeating the experiments on the second dataset, *Reuters_299_mixed*, using the parameter values found previously. We examine the results returned by our method, and evaluate whether it is suitable for datasets that contain documents that overlap across topics.

5.3.1 Experiment Settings

As the dataset contained documents tagged with more than one topic, certain adaptations were made in evaluating the quality of an equivalence class. For supporting documents that were tagged with two topics, we treated them as supporting both topics. For example, in an equivalence class containing the MFSs "*gas reserves*" and "*oil reserves*", 15 of the 21 supporting documents had two topic tags attached to them. Figure 5.1 contains the breakdown of the supporting documents for this equivalence class.

Breakdown of supporting documents for equivalence class
made up of MFSs "*gas reserves*" and "*oil reserves*"

Total number of supporting docs : 21

Among supporting docs,
number of docs tagged with topic "*crude*" only : 1
number of docs tagged with topic "*nat-gas*" only : 5
number of docs tagged with both topics : 15
number of docs with any other tags : 0

Figure 5.1. Breakdown of supporting documents for example equivalence class

In evaluating the quality of this equivalence class, we adopted the criterion as specified in Definition 4.1, i.e. at least 0.7 of the documents had to be from the same topic, for the class to be representative of that topic. The calculations in Figure 5.2 continue the analysis of this equivalence class, and we see that this class is representative of both topics "*crude*" and "*nat-gas*".

Percentage of docs supporting topic "*crude*" = (1+15)/21 = 0.761 > 0.7
Percentage of docs supporting topic "*nat-gas*" = (5+15)/21 = 0.952 > 0.7

$\therefore$, the equivalence class is a good class, and representative of BOTH topics.

Figure 5.2. Evaluation of quality of example equivalence class

For *Reuters_299_mixed*, we followed the same procedure as illustrated in Figure 4.1 in Chapter 4, and re-used these parameter values/ranges: a maximal word gap of 2, support from 9 to 5, and confidence from 0.8 to 0.5.

5.3.2 Experimental Results and Discussion

Table 5.4 shows the number of topics represented for each combination of support and confidence that we used, as well as the breakdown of topics that were found at each level of support. Our method was able to detect all five distinct topics in the dataset.

Table 5.4. Topic spread of equivalence classes from support 9 to 5, confidence 0.8 to 0.5, for *Reuters_299_mixed*

Confidence	Number of topics present				
	support 9	support 8	support 7	support 6	support 5
0.8	2	0	2	2	3
0.7	3	2	3	3	3
0.6	3	1	5	3	3
0.5	2	2	5	3	4
Topics found in confidence 0.8 to 0.5	1,2,4	2,4	1,2,3,4,5	2,4,5	1,2,3,4,5

The simple heuristic of grouping equivalence classes was used to group the classes into topic clusters. Table 5.5 contains a sample of the grouped equivalence classes from the *Reuters_299_mixed* dataset. The word phrases in the grouped equivalence classes act as topic descriptors for the topic they represent.

Table 5.5. Sample equivalence classes from each topic in *Reuters_299_mixed*

Equivalence Class 1	**Equivalence Class 2**	**Actual Topic**
alcan ltd, aluminium ltd	agreement mln, aluminum company	alum
oil barrel, west texas intermediate	oil prices dlrs, prices dlrs barrel	crude
grain elevator, trade sources	grain ships loading waiting load, loading ships	grain
cubic feet gas day, mln cubic feet day	natural gas reserves, reserves cubic feet	nat-gas
seamen strike, strike union spokesman	grain ships loading waiting load, loading ships	ship

Referring to highlighted entries in Table 5.5, it is seen that the equivalence class "*grain ships loading waiting load, loading ships*" can be categorized under the topic "*grain*" and "*ship*", since both contain elements that can be found in either topic. Upon checking the supporting documents, it is found that the equivalence class is indeed representative of both topics. Thus, in datasets where documents overlap different topics, the equivalence classes may also display the overlapping concepts, and are supported by the very documents that are associated with this overlap. This is useful if we want to detect which topics overlap, and if we want to find a representative set of documents that exhibit this topic-overlapping characteristic.

Table 5.6 shows the number of good and bad equivalence classes generated at each combination of support and confidence. The number of bad equivalence classes is once again a small percentage of the good equivalence classes, and so, the presence of the bad classes is not a major issue; the grouping of the good classes into clusters is still relatively unhindered.

Table 5.6. Percentage of good equivalence classes, from support 9 to 5, confidence 0.8 to 0.5, for *Reuters_299_mixed*

Support 9					Support 6				
confidence	No. of good classes	No. of bad classes	Total no. of classes	% good	confidence	No. of good classes	No. of bad classes	Total no. of classes	% good
0.8	2	0	2	100	0.8	4	0	4	100
0.7	3	0	3	100	0.7	8	0	8	100
0.6	2	0	2	100	0.6	6	0	6	100
0.5	2	0	2	100	0.5	8	0	8	100
Support 8					Support 5				
confidence	No. of good classes	No. of bad classes	Total no. of classes	% good	confidence	No. of good classes	No. of bad classes	Total no. of classes	% good
0.8	0	0	0	n.a.	0.8	6	2	8	75.0
0.7	1	0	1	100	0.7	8	2	10	80.0
0.6	1	0	1	100	0.6	8	2	10	80.0
0.5	3	0	3	100	0.5	12	1	13	92.3
Support 7									
Confidence	No. of good classes	No. of bad classes	Total no. of classes	% good					
0.8	3	0	3	100					
0.7	6	0	6	100					
0.6	7	0	7	100					
0.5	8	0	8	100					

Thus, by using the guideline values for support and confidence, we see that the five distinct topics of *Reuters_299_mixed* can be picked up, from the grouped equivalence classes that our method returns.

CHAPTER 6 – EXPERIMENTAL RESULTS FOR MANUFACTURING CORPUS VERSION 1 (MCV1)

In this section, we present the results of applying our hybrid method to subsets of the Manufacturing Corpus Version 1 (MCV1), which was originally assembled by Ying Liu et el. [6]. Through our experiments, we aim to determine whether our hybrid method is suitable for topic detection for a dataset of technical papers, and to also determine at which level (granularity based on the existing taxonomy of MCV1) of topic detection our method performs best.

In implementing our hybrid method, we re-used the guideline parameter values used in Chapter 5, for the Reuters-21578 datasets, namely:

- varying support from 9 to 5, and
- varying confidence from 0.8 to 0.5, and
- a maximal word gap of 2.

The heuristic that was previously used in the grouping of equivalence classes was dropped for the MCV1 dataset, because in the results, there were many topics that were represented by only one equivalence class, amongst a number of erroneous classes. The decision of matching equivalence classes to distinct topics in the dataset had to be done manually, instead of being a semi-automated task as originally proposed and implemented in the Reuters datasets.

6.1 MCV1 DATASET

MCV1 is an archive of 1,434 English language Manufacturing related engineering papers. It combines all engineering technical papers from Society of Manufacturing Engineers (SME) from year 1998 to year 2000 [6]. The corpus was assembled by Ying Liu and his associates, in an attempt to meet the needs of Manufacturing R&D

personnel who would often need to refer to well-classified technical literature in the course of their work. The coding process used in MCV1 differs markedly from other corpora, for example, the Reuters Corpus Volume 1 (RCV1), which consists of 800,000 Reuters newswire articles from 20/08/1996 to 19/08/1997 [24, 25]. In the coding for RCV1, more than 90 editors were involved, and the coding process was serial, e.g. a document is coded by an operator and subsequently checked by another. In MCV1, to take into account the constraints of labor and time, and to better track the subjectivity involved for each coding operator, an innovative parallel coding process was developed, requiring only four to eight operators. The coding process is broken into different phases. In the initial phase, the operators independently code the documents without any bias from each other's codes; in later phases, the operators compare and re-adjust their assigned codes through discussion [6]. This coding process ensures the final codes assigned are a result of a combined input and agreement amongst the coding operators, giving users a well-classified dataset with Manufacturing content.

The documents in MCV1 have been classified under 18 broad topics; these 18 topics are of low granularity, and within each topic, finer sub-topics may be found. The sub-topics found in the topic-tagging scheme can be up to two levels below than their original parent topic. This is illustrated in Figure 6.1.

Code	Topic Name
…	…
C02	2. Composites Manufacturing
C0201	1. Composites Manufacturing Fundamentals
C0202	2. Composites, Bonding & Joining
….	…
C0206	6. Composites Layup Processes
C020601	1. Filament Winding
C020602	2. Pultrusion
C020603	3. Resin Transfer Molding (RTM)
…	…

Figure 6.1. Example of taxonomy used in MCV1

In this report, we term the 18 parent topics as level-1 topics (e.g. C02), and the subsequent child topics as level-2 topics (e.g. C0202); similarly, the sub-topics of level-2 topics are termed level-3 topics (e.g. C020602). The full list of topics is found in Appendix C. This taxonomy of topics is adapted from that used by SME for the Manufacturing industry.

In our experiments, we only used the abstract of each document to represent the document itself. In the subsequent sections in this chapter, we will briefly describe the characteristics of each subset of MCV1 that was used in the experimentation, before presenting the results of applying our hybrid method to that subset.

In order to determine at which level of topic detection our method performs best, we generated subsets of MCV1 that were grouped by level-1, level-2 and level-3 topics, and evaluated the performance of our method on each of these subsets.

6.2 TOPIC DETECTION WITH LEVEL-1 TOPICS – *abstracts_146*

6.2.1 Dataset Characteristics

abstracts_146 is a subset of MCV1 we assembled, made up of 146 documents, across a range of seven topics. The documents were grouped according to their level-1 parent topics. This means that documents within the same topic might have different exact topic labels, but they share the same parent topic. The assembly details of *abstracts_146* can be found in Appendix D. This dataset was assembled to see if our method is able to detect topics that only generally describe the documents, since the topic labels used for these 146 documents are the parent topic labels and not the exact ones.

6.2.2 Experimental Results

In our method, the basic building blocks for forming topic clusters are MFSs. For *abstracts_146*, using the guideline parameters as stated in Section 6.1, there were very few MFSs returned. The consequence was that there were no returned equivalence classes. Thereafter, we extended the range of values for confidence; across 0.9 to 0.1. This was done to see if the non-performance of the method was an issue of the tweaking of parameters. The results (for number of equivalence classes returned) are summed up in Table 6.1.

Table 6.1. Number of equivalence classes returned, at various parameter combinations, for *abstracts_146*

	Number of equivalence classes returned				
Confidence	support 9	support 8	support 7	support 6	support 5
0.9	0	0	0	0	0
0.8	0	0	0	0	0
0.7	0	0	0	0	0
0.6	0	0	0	0	0
0.5	0	0	0	0	0
0.4	0	0	0	0	0
0.3	0	0	0	0	0
0.2	0	0	0	0	1
0.1	0	0	1	1	0

The boxed-up area in Table 6.1 reflects that no equivalence classes were returned when the guideline parameters were used. Exploring the results beyond the guideline parameters, it is observed that equivalence classes are returned for certain parameter combinations. However, the equivalence classes found are not representative (as defined in Definition 4.1) of the topics in the dataset.

For example, for support 7, confidence 0.1, the single returned equivalence class contained the following MFSs:

- *manufacturing systems*
- *paper describes*

Upon examining the supporting documents of these 2 MFSs, it is found that the documents come from six different topics, with no one dominant topic. This phenomenon is repeated amongst the other equivalence classes that were found. Hence, we are unable to generate topic clusters that cover all seven topics in the *abstracts_146* dataset.

6.3 TOPIC DETECTION WITH LEVEL-3 TOPICS – *abstracts_349* and *abstracts_252*

6.3.1 Dataset Characteristics

abstracts_349 is made up of 349 documents, across a range of seven topics. The documents were grouped according to their specific level-3 topics. Documents within the same topic share the exact same topic label, since a level-3 topic is of the finest granularity in the coding policy used. The assembly details of *abstracts_349* can be found in Appendix D.

abstracts_252 is a subset of *abstracts_349*, and is made up of 252 documents, across a range of five topics. The difference in these two datasets was that two of the topics from *abstracts_349* were removed. These two topics were conceptually close to another two of the existing topics; they shared the same level-2 parent topic. The assembly details of *abstracts_252* can be found in Appendix D.

We assembled both *abstracts_349* and *abstracts_252* to investigate the performance of our method in detecting topics that match document content very specifically. A resulting property of these two datasets is a high similarity of document content within a topic.

6.3.2 Experimental Results

We applied the topic detection method on *abstracts_349* first, and later on *abstracts_252*, to compare the effect of having conceptually close level-3 topics within a dataset.

For *abstracts_349*, we used the guideline parameters as stated in Section 6.1, and managed to detect five out of the seven topics amongst the equivalence classes returned. We extended the range of confidence used once again, and managed to pick up the remaining two topics in the additional equivalence classes generated. Table 6.2 sums up the topic spread of the equivalence classes generated, across the parameters we used.

Table 6.2. Topic spread of equivalence classes from support 9 to 5, confidence 0.9 to 0.1, for *abstracts_349*

	Number of topics present				
Confidence	support 9	support 8	support 7	support 6	support 5
0.9	0	0	0	0	0
0.8	0	0	0	0	1
0.7	1	0	0	0	1
0.6	1	0	1	0	1
0.5	1	0	2	0	2
0.4	1	0	1	2	0
0.3	1	1	0	0	0
0.2	0	0	0	0	0
0.1	0	0	0	0	0
Topics found in confidence 0.8 to 0.5	4	-	5,6	-	1,3
Topics found outside of guideline parameters	4	4	5	2,7	-

However, the good equivalence classes are not easily detected, because at each parameter combination used, a fair number of bad equivalence classes are generated as well. Table 6.3, which shows the percentage of good equivalence classes for each parameter combination (the '-' indicates that no equivalence classes were generated for that particular combination), illustrates this point, where most of the percentages fall below 50%. The significant presence of these bad equivalence classes poses an

issue, considering that we are manually choosing the equivalence classes that will represent the distinct topics in the MCV1 datasets.

Table 6.3. Percentage of good equivalence classes, from support 9 to 5, confidence 0.9 to 0.1, for *abstracts_349*

	Percentage of good equivalence classes (%)				
Confidence	support 9	support 8	support 7	support 6	support 5
0.9	-	-	-	-	-
0.8	-	-	-	-	33.3
0.7	100.0	-	-	-	33.3
0.6	100.0	-	100.0	0.0	14.3
0.5	50.0	-	40.0	0.0	16.7
0.4	50.0	-	25.0	28.6	0.0
0.3	100.0	33.3	0.0	0.0	0.0
0.2	-	-	-	0.0	-
0.1	-	-	-	-	-

We repeated the experimental procedure on *abstracts_252*, and the figures for the topic spread are seen in Table 6.4.

Table 6.4. Topic spread of equivalence classes from support 9 to 5, confidence 0.9 to 0.1, for *abstracts_252*

	Number of topics present				
Confidence	support 9	support 8	support 7	support 6	support 5
0.9	0	0	0	0	0
0.8	0	0	0	0	1
0.7	1	0	0	0	1
0.6	1	1	1	0	1
0.5	1	1	1	0	2
0.4	2	1	1	1	1
0.3	2	3	0	0	0
0.2	1	1	1	0	0
0.1	0	0	0	0	1
Topics found in confidence 0.8 to 0.5	3	4	4	-	1,2,5
Topics found outside of guideline parameters	2,3,4	2,3,4	2,4	2	1,2

From Table 6.4, it is observed that all five topics in the dataset are detected, but not within the guideline parameters. The issue of bad equivalence classes has improved from the situation seen in *abstracts_349*. Out of the parameter combinations that did produce equivalence classes, most of them generated classes that were representative of the topics in the dataset. This is seen in Table 6.5, where most of the percentage figures are 50% and above.

Table 6.5. Percentage of good equivalence classes, from support 9 to 5, confidence 0.9 to 0.1, for *abstracts_252*

	Percentage of good equivalence classes (%)				
Confidence	support 9	support 8	support 7	support 6	support 5
0.9	-	-	-	-	-
0.8	-	-	-	-	50.0
0.7	100.0	-	-	-	50.0
0.6	100.0	100.0	50.0	0.0	25.0
0.5	100.0	100.0	50.0	0.0	40.0
0.4	100.0	66.7	60.0	40.0	25.0
0.3	100.0	100.0	0.0	0.0	0.0
0.2	100.0	100.0	100.0	-	-
0.1	-	-	-	-	100.0

6.4 TOPIC DETECTION WITH LEVEL-2 TOPICS – *abstracts_160* and *abstracts_197*

6.4.1 Dataset Characteristics

abstracts_160 is made up of 160 documents, across a range of five topics. The documents were grouped according to their specific level-2 topics. The documents, when grouped to their respective topics, share the exact same level-2 topic label. They are not grouped together based on a parent topic, which was the case for *abstracts_146*, the dataset used in Section 6.2. The assembly details of *abstracts_160* can be found in Appendix D.

abstracts_197 is similar to *abtracts_160*; it is made up of 197 documents, across a range of a different set of five level-2 topics. Once again, the documents within each topic share the exact same level-2 topic label. The assembly details of *abstracts_160* can be found in Appendix D.

We assembled both *abstracts_160* and *abstracts_197* to investigate the performance of our method in detecting topics that match document content, but not as specifically as the datasets in Section 6.3. The results from these two datasets can then be compared with that of the level-3 topic datasets, to see how closeness of document content within a topic affects topic detection using our method.

6.4.2 Experimental Results

We applied the topic detection method on *abstracts_160*, using both the guideline parameters and the additional range of confidence values. The results we got are summed up in Table 6.6.

Table 6.6. Topic spread of equivalence classes from support 9 to 5, confidence 0.9 to 0.1, for *abstracts_160*

	Number of topics present				
Confidence	support 9	support 8	support 7	support 6	support 5
0.9	0	0	0	0	0
0.8	0	0	0	0	2
0.7	0	0	0	1	2
0.6	0	0	1	2	1
0.5	0	0	1	0	2
0.4	1	1	2	1	2
0.3	0	1	1	1	2
0.2	0	2	1	1	1
0.1	1	0	1	0	0
Topics found in confidence 0.8 to 0.5	-	-	2	1,5	1,5
Topics found outside of guideline parameters	1	1,2,5	1,2,5	1,5	1,5

From Table 6.6, we see that even with the use of the additional range of confidence values, the coverage of topics is not wide enough to detect all five topics. Only three of the five topics are picked up in the generated equivalence classes.

We repeated the experiments on *abstracts_197,* and our topic-spread results are summed up in Table 6.7.

Table 6.7. Topic spread of equivalence classes from support 9 to 5, confidence 0.9 to 0.1, for *abstracts_197*

	Number of topics present				
Confidence	support 9	support 8	support 7	support 6	support 5
0.9	0	0	0	0	0
0.8	0	0	0	0	0
0.7	0	0	0	0	1
0.6	0	0	0	0	2
0.5	0	0	0	1	2
0.4	0	0	0	1	1
0.3	0	0	0	0	1
0.2	0	0	0	0	0
0.1	0	0	0	0	0
Topics found in confidence 0.8 to 0.5	-	-	-	2	1,2
Topics found outside of guideline parameters	-	-	-	2	2

From Table 6.7, we see that the coverage of topics is once again not wide enough to detect all five topics. Only two topics are picked up.

6.5 DISCUSSION OF RESULTS

Comparing the results from the different datasets, it is observed that the datasets with the level-3 topics, *abstracts_349* and *abstracts_252*, were more suitable for our topic detection method; all the topics in these two datasets were detected, with each topic having at least one equivalence class to represent it. This suggests that our method is better able to detect the distinct topics in a dataset if the topics in question are very specific and well defined; this also implies the document content in each topic exhibit a high degree of similarity to each other.

An explanation for this observation can be made, if we examine the characteristics of each of the datasets. In the level-1 topic dataset *abstracts_146*, the matching between document content and topic labels was poor, since the topic labels we used for each document were the level-1 parent topics. It is highly likely there was little similarity between documents grouped in the same topic. Our method relies on the presence of MFSs as an indicator that documents are similar in content. However, with dissimilar documents grouped under the same topic, the number of supporting documents for some of the topic-representative word sequences might not even meet the support threshold. Only word sequences that appear across all topics, and are applicable to Manufacturing research in general, would be detected. This is reflected in the results for *abstracts_146* in Section 6.2.2, where the MFSs "*manufacturing systems*" and "*paper describes*" were picked up by our method.

Continuing our examination of dataset characteristics with the level-3 topic datasets, we see that the documents within the same topic are highly similar in content. This high similarity possibly translates into many topic-representative word sequences that

are repeated across a sufficient number of documents. This accounts for good topic coverage in the level-3 sets, which we see in Section 6.3.2. For the level-2 topic datasets, document content within the same topic is similar, but the topic specification for a set of grouped documents is not as well defined as that of the level-3 sets. This possibly has the effect of much fewer topic-representative frequent sequences being detected, as compared to the level-3 sets, which may result in some topics not being represented. This is seen to be true in our results in Section 6.4.2, where topic-representative word sequences that are frequent are still detected in the level-2 sets, but topic coverage is incomplete.

We also see that with regards to the confidence parameter, the guideline values that we arrived at for the Reuters-21578 datasets are not necessarily the best values to use for the MCV1 datasets. This implies that the ideal range of values for the confidence parameter is dataset-dependent, and prior experimentation needs to be done with a sample of pre-classified documents from the dataset, to determine these values.

The issue of bad equivalence classes being generated warrants some discussion. Comparing the results between *abstracts_349* and *abstracts_252*, it is seen in Table 6.5 that bad equivalence classes took up a smaller proportion of the classes generated across all parameter combinations, for the latter dataset. This suggests that for datasets similar to the ones we used in this chapter – single-topic technical documents – the topic detection method performs better when the topics in the dataset are conceptually well separated from each other. This would translate into the supporting documents of a MFS coming from one distinct topic, instead of a few topics, and the number of bad equivalence classes would thus be reduced.

This seems to be less of an issue for the Reuters datasets. Upon comparing the percentage figures in Table 5.3 and 5.6, with that of Table 6.5 and 6.7, it is seen that bad equivalence classes in the Reuters datasets, in general, make up a smaller proportion of the returned equivalence classes. We could attribute this to the different nature of the Reuters and MCV1 datasets. The Reuters documents are newswire articles, and the topics dealt with are in themselves distinct and conceptually separated from each other. The MCV1 documents are technical articles, and already have an underlying association with Manufacturing. Therefore, they are conceptually related in some sense. Technical terms within the field of Manufacturing may be used across topics in the dataset, and this would lead to MFSs (and eventually, equivalence classes) having supporting documents that come from a range of topics, instead of one distinct and dominant topic.

CHAPTER 7 – APPLICATION OF TOPIC DETECTION METHOD ON *abstracts_319*

Our observations in the previous chapters show that our method is better suited to datasets where the distinct topics are well defined and conceptually separated from each other. However, in practice, real-world datasets seldom come in such ideal conditions. Instead, there is usually a mix of all the dataset characteristics that we have tried to isolate and deal with independently so far; there would be both single and multi-topic documents, and documents that are related to each other with a varying degree of content similarity. In light of this, we want to explore the usefulness of our topic detection method on a dataset that approximates a real-world Manufacturing dataset.

In this section, we present the results of applying our hybrid method on a subset of MCV1 we assembled, *abstracts_319*. This dataset is made up of the abstracts of papers in MCV1, from year 1998 only. This dataset simulates a real world collection of technical articles, which come from a wide range of topics, which we have no prior knowledge of. We aim to make certain conclusions about the identification of distinct topics in this dataset, through the use of our topic detection method.

7.1 DATASET CHARACTERISTICS

Based on the classification that was done by Ying Liu et el. [6], the documents in MCV1 from year 1998 span 196 distinct topics, out of which only 53 topics have at least 5 documents supporting them. The topics found in *abstracts_319* include level-1 to level-3 topics. Amongst the 319 documents in the dataset, some are singly tagged, whereas others have multiple topic assignments.

7.2 EXPERIMENTAL PROCEDURE

The initial procedure of using the method is the same; we run the topic detection method, using a support range of 9 to 5, a confidence range of 0.9 to 0.1, and a maximal word gap of 2. The heuristic of grouping equivalence classes together is once again not used, for the same reasons outlined in Chapter 6. We then examine the returned equivalence classes, and see whether the MFSs contained within are descriptive of topics related to Manufacturing. This is will be done using our background in Manufacturing.

We do not restrict ourselves to the performance measures of wide topic spread and proportion of good equivalence classes, which were used in all the previous datasets. After grouping documents into a topic cluster, we will also qualitatively examine whether the original documents really have some common underlying topic, and whether the MFSs accurately reflect the topic.

7.3 EXPERIMENTAL RESULTS

We ran the topic detection method, using a support range of 9 to 5, a confidence range of 0.9 to 0.1, and a maximal word gap of 2.

Upon examining the returned equivalence classes, we first remove the equivalence classes that we think will not be useful in characterizing a distinct topic in the dataset. An example of this is an equivalence class that contains these 2 MFSs:

- *planning system*
- *system planning*

We next group the remaining classes into topics that they represent. From the results, we managed to group them into 7 main topics. The full list, including our interpretation of a probable topic for each topic cluster, is found in Appendix E. We take a closer look at some of the equivalence classes that were identified.

Class 5:

- *product development*
- *concurrent engineering*

There are 10 supporting documents from this equivalence class. The original topic assignments do not exhibit any particular pattern, but upon examination of the abstracts of these documents, we see that they contain, to some extent, content that has to do with concurrent engineering and shortening production time. This is captured by the MFSs in the cluster.

Class 7:

- *tool life*
- *cutting edge*

There are 10 supporting documents for this equivalence class. Upon inspection of the topic assignments for each of these documents, we discover that 9 of the documents are tagged with at least 1 topic that has “C07 – Machining and Material Removal Processes” as its parent topic. The 2 MFSs above reflect one aspect of this parent topic, and in that sense, they are descriptive of the distinct topic that is common to the documents within the cluster.

7.4 DISCUSSION OF RESULTS

From the closer examination of the equivalence classes in Section 7.3, we see that when presented with a dataset of varied and overlapping topics, the topic detection method is able to identify certain major topics. This may or may not agree with the existing classification that might already be in place, because the classification is after all also a process which involves subjective human decisions; our method is just as likely to group documents in a manner that has not been picked up by the domain experts who provided the original classification in the first place.

The usefulness of the method, therefore, would be to provide an initial identification of topics in *abstracts_319*. This list of topics is by no means exhaustive; it would not be possible to detect the 196 topics that the documents were originally tagged with, using our method. However, the 7 main topics we arrived at serves as an intermediate result to further study and partition the dataset.

Our topic detection method may then be used in a same manner for other datasets which we have some domain knowledge about, and possibly spanning a wide range of topics. The results would similarly serve as an intermediate result to work upon.

CHAPTER 8 – CONCLUSION AND FUTURE WORK

8.1 CONCLUSION

The topic detection method we have developed and implemented is a hybrid of the existing MFS Discovery algorithm and a heuristic to group MFSs into topic clusters. To detect distinct topics in a textual dataset, we adapted Ahonen's [19, 20] work in MFS Discovery and equivalence class generation, by performing MFS Discovery and equivalence class generation across a range of support and confidence values, respectively. From the set of returned equivalence classes, we used a heuristic to group classes into topic clusters.

Based on the results of applying our method on the different datasets, we can arrive at the following conclusions:

1. The method is useful in detecting distinct topics in a dataset where the topics are well separated in terms of subject matter. Even if some of the topics overlap across documents, the method is still able to fulfill its purpose. This is seen in the Reuters-21578 datasets, where the presence of bad equivalence classes is not a big issue, and our method was able to detect topics even in a dataset where some documents had multiple topic assignments. The MFSs in the returned equivalence classes also served their purpose of describing the topics that they represent.
2. For datasets whose documents all have an underlying association with each other to begin with, the method is still able to work, but does not measure up to the performance seen in the datasets as described in the previous point. Also, the grouping of equivalence classes according to topic has to be done

manually. For these datasets that are identified by some common subject, the method favors datasets whose topics are very specific in nature, and conceptually well separated from each other. This is seen in the MCV1 datasets, where the level-3 topic dataset (that did not have any common level-2 parent topics amongst its topics) exhibited the best performance.

3. For real world datasets that are not so simply defined in terms of the topics present, our method serves as a way to generate a list of distinct topics which will by no means be exhaustive or fully descriptive of the topics present, but it will instead act as an intermediate result to understanding and further partitioning the dataset. Some domain knowledge will be required, in deciding which equivalence classes are representative of the topics. The method was used in this manner for the dataset *abstracts_319*.

8.2 FUTURE WORK

An area in which our method can be improved is in the returned MFSs themselves. As mentioned in Chapter 6, in our discussion of results, our method relies on the presence of MFSs as an indicator that documents are similar in content. In reality, however, we also generate MFSs that span across all topics in the dataset, leaving us with many MFSs that are not useful in distinguishing the distinct topics. One way to deal with this is to use some statistical measure, to determine the significance of each MFS, and only select the top-ranking *k* MFSs. A similar step was undertaken in [26], where Hirao et el. used a χ^2 test to narrow down the number of sequential word patterns they extracted out of a dataset, based on the statistical significance of each pattern. Removing the irrelevant MFSs early into the method would help us avoid subsequent

problems of equivalence classes not being representative of topics, since equivalence classes are made up of MFSs.

Our heuristic to group equivalence classes into topic clusters is based on the assumption that equivalence classes of the same topic would have co-occurring words. While this is true for some of the equivalence classes in the Reuters datasets, there are several exceptions; also, it was inappropriate to use the heuristic for the MCV1 datasets. For the case of the Reuters datasets, the confusion arises when there are co-occurring words between equivalence classes of different topics. Upon examination, these words are words that do not form the main subject of the equivalence class they occur in; they are instead used in conjunction with words that do distinguish one topic from another. As such, the heuristic can be improved for the Reuters datasets (and possibly other datasets whose topics are conceptually well separated) by using a statistical measure to determine whether a particular co-occurring word is a keyword in a topic. For the MCV1 datasets, the greater issue is the elimination of the bad equivalence classes amongst the good ones; the heuristic would not be useful with the returned results in any case, because there was usually only 1 equivalence class representing each topic, i.e. there is no second or more equivalence classes that need to be grouped together with a current one, to form a topic cluster. One way to tackle the bad clusters has already been mentioned, and that is to tackle the problem at the source, by eliminating irrelevant MFSs.

Another avenue to build upon the work in this thesis is to consider how else to use MFSs in topic detection, as given in the title of this work. One possible way would be to modify the work of Fung et el. [27], in which frequent itemsets were used to

perform hierarchical document clustering. In the context of Fung's work, a frequent itemset refers to a collection of words in the textual dataset that co-occur in at least a minimum fraction of the whole dataset. MFSs differ from frequent itemsets in that an MFS is maximally frequent (it does not have a frequent super-set) and the order of words is taken into account. The effect of replacing the frequent itemsets with MFSs would be an interesting research topic, to see how taking the sequence into account would affect the existing clustering accuracy (of [27]), and whether this hybrid method outperforms our proposed method in this current work.

REFERENCES

[1] E. Simoudis, Reality check for data mining, IEEE Expert: Intelligent Systems and Their Applications, 11(5) (1996), pp. 26-33.

[2] A. Strehl, J. Ghosh, R. Mooney, Impact of similarity measures on web-page clustering, in: Proceedings of AAAI Workshop on AI for Web Search (2000), pp. 58-64.

[3] Y. Seo, K. Sycara, Text clustering for topic detection, Technical report CMU-RI-TR-04-03, Robotics Institute, Carnegie Mellon University, January, 2004.

[4] F. Walls, H. Jin, S. Sista, R. Schwartz, Topic detection in broadcast news, in: Proceedings of the DARPA Broadcast News Workshop, pp. 193-198, 1999.

[5] Reuters-21578 News Collection. Available from: http://www.daviddlewis.com/resources/testcollections/reuters21578/reuters21578.tar.gz

[6] Y. Liu, H.T. Loh, S.B. Tor, Building a document corpus for Manufacturing knowledge retrieval, in: Singapore MIT Alliance Symposium 2004, Singapore, 2004.

[7] J. Allan, J. Carbonell, G. Doddington, J. Yamron, Y. Yang, Topic detection and tracking pilot study: Final report, in: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, Virginia, 1998, pp. 194-218.

[8] E. Bingham, A. Kaban, M. Girolami, Topic identification in dynamical text by complexity pursuit, Neural Processing Letters 17(1) (2003), pp. 69-83.

[9] K. Aas, L. Eikvil,, 1999. Text categorization: A survey, Technical report, Norwegian Computing Center, June 1999.

[10] J. Han, M. Kamber, Data mining: Concepts and techniques. Morgan Kaufmann Publishers, New York, 2001.

[11] G. Salton, M.J. McGill, Introduction to modern retrieval, McGraw-Hill Book Company, 1983.

[12] L. Kaufman, R.J. Rousseeuw, Finding groups in data – An introduction to cluster analysis, John Wiley & Sons, New York, 1990.

[13] G.W. Miligan, Clustering validation: Results and implications for applied analyses, in: P. Araboe, L.J. Hubert, G. de Soete (Eds.), Clustering and Classification, World Scientific Publishing, Singapore, 1996, pp. 341–373.

[14] P. Willett, Recent trends in hierarchical document clustering: A critical review, Information Processing & Management 24 (5) (1988), pp. 577-597.

[15] R.A. Johnson, D.W. Wichern, Applied multivariate statistical analysis, Prentice Hall, New Jersey, 2002.

[16] Y. Zhao, G. Karypis, Evaluation of hierarchical clustering algorithms, for document datasets, ACM Press, 2002, pp. 515-524.

[17] M. Rorvig, Images of similarity: A visual exploration of optimal similarity metrics and scaling properties of TREC topic-document sets, Journal of the American Society for Information Science, 50(8) (1999), pp. 639-651.

[18] F. Meziane, Y. Rezgui, A document management methodology based on similarity contents, Information Sciences 158 (2004), pp. 15-36.

[19] H. Ahonen, Finding all maximal frequent sequences in text, in: Proceedings of ICML-99 Workshop, Machine Learning in Text Data Analysis, Bled, Slovenia, 1999.

[20] H. Ahonen-Myka, O. Heinonen, M. Klemettinen, A.I. Verkamo, Finding co-occurring text phrases by combining sequence and frequent set discovery, in: R. Feldman (Ed.), Proceedings of 16th International Joint Conference on Artificial Intelligence IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications, Stockholm, Sweden, 1999, pp. 1-9.

[21] R. Agrawal, T. Imielinski, A. Swami, Mining association rules between sets of items in large databases, in: Proceedings of 1993 ACM SIGMOD Conference, Washington DC, USA, 1993, pp. 207-216.

[22] H. Mannila, H. Toivonen, A.I. Verkamo, Efficient algorithms for discovering association rules, in: Proceedings of AAAI Workshop Knowledge Discovery in Databases, Seattle, Washington, 1994, pp. 181-192.

[23] Onix Text Retrieval Toolkit: Stop Word List 1. Available from http://www.lextek.com/manuals/onix/stopwords1.html.

[24] D.D. Lewis, Y. Yang, T.G. Rose, F. Li, RCV1: A new benchmark collection for text categorization research, The Journal of Machine Learning Research, 5 (2004), pp. 361-397.

[25] T. Rose, M. Stevenson, M. Whitehead, The Reuters Corpus Volume 1 – From yesterday's news to tomorrow's language resources, presented at 3rd international conference on language resource and evaluation, 2002.

[26] T. Hirao, J. Suzuki, H. Isozaki, E. Maeda, NTT's multiple document summarization system for DUC2004, in: Proceedings of Document Understanding Conference 2004, Boston, USA, 2004.

[27] B. Fung, K. Wang, M. Ester, Hierarchical document clustering using frequent itemsets, in: Proceedings of the SIAM International Conference on Data Mining (2003).

APPENDIX A – TOPIC LIST IN REUTERS-21578

The documents in Reuters-21578 have been tagged according to a list of 135 topics, which deal mostly with business and the economy. These topic codes are found between the <TOPIC> meta-tags of each document. The list of topics is given below.

```
****Subject Codes (135)

Money/Foreign Exchange (MONEY-FX)
Shipping (SHIP)
Interest Rates (INTEREST)

**Economic Indicator Codes (16)

Balance of Payments (BOP)
Trade (TRADE)
Consumer Price Index (CPI)
Wholesale Price Index (WPI)
Unemployment (JOBS)
Industrial Production Index (IPI)
Capacity Utilisation (CPU)
Gross National/Domestic Product (GNP)
Money Supply (MONEY-SUPPLY)
Reserves (RESERVES)
Leading Economic Indicators (LEI)
Housing Starts (HOUSING)
Personal Income (INCOME)
Inventories (INVENTORIES)
Instalment Debt/Consumer Credit (INSTAL-DEBT)
Retail Sales (RETAIL)

**Currency Codes (27)

U.S. Dollar (DLR)
Australian Dollar (AUSTDLR)
Hong Kong Dollar (HK)
Singapore Dollar (SINGDLR)
New Zealand Dollar (NZDLR)
Canadian Dollar (CAN)
Sterling (STG)
D-Mark (DMK)
Japanese Yen (YEN)
Swiss Franc (SFR)
French Franc (FFR)
Belgian Franc (BFR)
Netherlands Guilder/Florin (DFL)
Italian Lira (LIT)
Danish Krone/Crown (DKR)
Norwegian Krone/Crown (NKR)
Swedish Krona/Crown (SKR)
Mexican Peso (MEXPESO)
Brazilian Cruzado (CRUZADO)
Argentine Austral (AUSTRAL)
Saudi Arabian Riyal (SAUDRIYAL)
South African Rand (RAND)
```

```
Indonesian Rupiah (RUPIAH)
Malaysian Ringitt (RINGGIT)
Portuguese Escudo (ESCUDO)
Spanish Peseta (PESETA)
Greek Drachma (DRACHMA)

**Corporate Codes (2)

Mergers/Acquisitions (ACQ)
Earnings and Earnings Forecasts (EARN)

**Commodity Codes (78)

ALUM
BARLEY
CARCASS
CASTOR-MEAL
CASTOR-OIL
CASTORSEED
CITRUSPULP
COCOA
COCONUT-OIL
COCONUT
COFFEE
COPPER
COPRA-CAKE
CORN-OIL
CORN
CORNGLUTENFEED
COTTON
COTTON-MEAL
COTTON-OIL
COTTONSEED
F-CATTLE
FISHMEAL
FLAXSEED
GOLD
GRAIN
GROUNDNUT
GROUNDNUT-MEAL
GROUNDNUT-OIL
IRON-STEEL
LEAD
LIN-MEAL
LIN-OIL
LINSEED
LIVESTOCK
L-CATTLE
HOG
LUMBER
LUPIN
MEAL-FEED
NICKEL
OAT
OILSEED
ORANGE
PALLADIUM
PALM-MEAL
PALM-OIL
PALMKERNEL
PLATINUM
```

```
PLYWOOD
PORK-BELLY
POTATO
RAPE-MEAL
RAPE-OIL
RAPESEED
RED-BEAN
RICE
RUBBER
RYE
SILK
SILVER
SORGHUM
SOY-MEAL
SOY-OIL
SOYBEAN
STRATEGIC-METAL
SUGAR
SUN-MEAL
SUN-OIL
SUNSEED
TAPIOCA
TEA
TIN
TUNG-OIL
TUNG
VEG-OIL
WHEAT
WOOL
ZINC

**Energy Codes (9)

Crude Oil (CRUDE)
Heating Oil/Gas Oil (HEAT)
Fuel Oil (FUEL)
Gasoline (GAS)
Natural Gas (NAT-GAS)
Petro-Chemicals (PET-CHEM)
Propane (PROPANE)
Jet and Kerosene (JET)
Naphtha (NAPHTHA)
```

APPENDIX B – ASSEMBLY OF REUTERS DATASETS

The assembly details of the *Reuters_261* and *Reuters_299_mixed* are presented here, to facilitate the verifying of results and subsequent research work with these datasets.

B.1. *Reuters_261*

This dataset consists of 6 topics:

- alum
- grain
- ipi
- iron-steel
- nat-gas
- reserves

The documents in the Reuters-21578 dataset are processed one by one; the <TOPIC> meta-tag of each document is checked to see if it contains only a single topic, and whether it is one of these 6 topics. If this is fulfilled, the <BODY> meta-tag is checked to see if there is a body of text in the document. Only documents that fulfill these 2 criteria are extracted. This assembly procedure gives us a dataset of 261 documents.

B.2. *Reuters_299_mixed*

This dataset consists of 5 topics:

- alum
- crude
- grain
- nat-gas
- ship

Within this dataset, we have documents that are:

- singly tagged with 1 of these 5 topics,
- tagged with both "*grain*" and "*ship*" topics, and
- tagged with both "*crude*" and "*nat-gas*" topics.

The purpose of introducing documents that are tagged with more than 1 topic is to simulate a real world document collection, which usually contains documents that span across a few topics.

The assembly of this dataset is similar to that described in B.1; the documents in the Reuters-21578 dataset are processed one-by-one, and only documents with an actual body of text are considered. Only documents whose topic tags fit the 3-point description given above are chosen. This assembly procedure gives us a dataset of 299 documents.

APPENDIX C – TOPIC-TAGGING SCHEME FOR MCV1

The 1,434 documents in MCV1 were classified according to a taxonomy of topics, that was adapted from that used by SME for the Manufacturing industry. The topic-tagging scheme is given here. There are 18 Level 1 topics in all (of the format C*XX*), excluding C19, which is used to tag documents that cannot be classified into any of the labels before it. A document in MCV1 may be tagged with more than one topic label.

CODE	TOPIC NAME
C01	1. Assembly & Joining
C0101	1. Adhesive Bonding
C0102	2. Assembly & Joining Fundamentals
C0103	3. Assembly Test & Inspection
C0104	4. Automated Assembly
C0105	5. Brazing
C0106	6. Composites Manufacturing
C0107	7. Fastening
C0108	8. Material & Part Handling for Assembly
C0109	9. Riveting
C0110	10. Soldering
C0111	11. Wire Processing
C01TH	12. Others
C02	2. Composites Manufacturing
C0201	1. Composites Manufacturing Fundamentals
C0202	2. Composites, Bonding & Joining
C0203	3. Composites, Part Sealing
C0204	4. Composites, Sheet Molding Compounds (SMC)
C0205	5. Composites, Tooling, Molds & Patterns
C0206	6. Composites Layup Processes
C020601	1. Filament Winding
C020602	2. Pultrusion
C020603	3. Resin Transfer Molding (RTM)
C020604	4. Spray-up
C0207	7. Composites, Curing Methods & Equipment
C020701	1. Composites, Electron Beam Curing
C020702	2. Composites, Oven Curing
C0208	8. Composites, Matrix Materials
C020801	1. Composites, Matrices, Carbon
C020802	2. Composites, Matrices, Ceramic
C020803	3. Composites, Matrices, Metals

CODE	TOPIC NAME
C020804	4. Composites, Matrices, Polymers
C02TH	9. Others
C03	3. Electronics Manufacturing
C0301	1. Electronics Manufacturing Fundamentals
C0302	2. Microelectronics Fabrication & Assembly
C03TH	3. Others
C04	4. Finishing & Coating
C0401	1. Finishes, Curing
C0402	2. Finishing & Coating Fundamentals
C0403	3. Material & Part Handling for Finishing
C0404	4. Parts Cleaning, Degreasing
C0405	5. Quality & Inspection of Finishes
C0406	6. Substrate Selection & Pretreatment
C0407	7. Coating Specific Substrates
C040701	1. Painting, Metal Substrates
C040702	2. Painting, Plastics Substrates
C040703	3. Painting, Wood Substrates
C0408	8. Finishing Processes
C040801	1. Anodizing
C040802	2. Automated Coating
C040803	3. Dip Coating
C040804	4. Electrocoating (E-Coat)
C040805	5. Electrostatic Finishing
C040806	6. Metallizing
C040807	7. Painting
C040808	8. Plating & Electroplating
C040809	9. Powder Coating Processes
C040810	10. Robotic Finishing
C040811	11. Spray Finishing
C040812	12. Vapor Deposition
C04TH	9. Others
C05	5. Forming & Fabricating
C0501	1. Coil Handling & Processing
C0502	2. Cold & Warm Forming
C0503	3. Extruding
C0504	4. Forging
C0505	5. Lubricants for Metal Forming
C0506	6. Metalforming Fundamentals
C0507	7. Part Handling, Metalforming
C0508	8. Press Feeding
C0509	9. Press Selection & Evaluation
C0510	10. Sheet Metal Formability Fundamentals
C0511	11. Thread Rolling

CODE	TOPIC NAME
C0512	12. Casting
C051201	1. Casting
C051202	2. Die Casting
C051203	3. Lost Foam Casting
C051204	4. Lost Wax & Investment Casting
C051205	5. Metal Mold Casting
C051206	6. Sand Mold Casting
C0513	13. Dies
C051301	1. Die Changing, Transfer & Handling
C051302	2. Die Design & Layout
C051303	3. Die Maintenance & Repair
C051304	4. Die Making
C051305	5. Die Materials
C0514	14. Sheet & Tube Forming
C051401	1. Bending
C051402	2. Drawing
C051403	3. Folding
C051404	4. Hydroforming
C051405	5. Roll Forming
C051406	6. Stamping
C051407	7. Stretch Forming
C0515	15. Sheet Metal Fabricating
C051501	1. Blanking
C051502	2. Fineblanking
C051503	3. Laser & Plasma Cutting
C051504	4. Nibbling & Notching
C051505	5. Plate & Structural Fabricating
C051506	6. Press Brakes
C051507	7. Punching
C051508	8. Shearing
C051509	9. Waterjet Cutting
C05TH	16. Others
C06	6. Lean Manufacturing & Supply Chain Management
C0601	1. Continuous Improvement
C0602	2. Just-in-Time (JIT)
C0603	3. Lead Time Reduction, Cycle Time Reduction
C0604	4. Lean Manufacturing Fundamentals
C0605	5. Lean Production
C0606	6. Mistake Proofing (Poka-Yoke)
C0607	7. Pull Systems (Kanban)
C0608	8. Quick Changeover
C0609	9. Supply Chain Management
C0610	10. Total Productive Maintenance (TPM)
C0611	11. Value Stream Analysis
C06TH	12. Others

CODE	TOPIC NAME
C07	7. Machining & Material Removal Processes
C0701	1. Boring
C0702	2. Broaching, Planing, Shaping, & Slotting
C0703	3. Cutting Fluids & Lubricants
C0704	4. Drilling, Reaming, Tapping & Related Processes
C0705	5. Dry Machining
C0706	6. Gear & Spline Processing
C0707	7. High Speed Machining
C0708	8. Machining Centers
C0709	9. Machining Composites
C0710	10. Micromachining
C0711	11. Milling
C0712	12. Sawing
C0713	13. Screw Machining
C0714	14. Threading
C0715	15. Turning
C0716	16. Workholding & Fixturing
C0717	17. Abrasive Machining
C071701	1. Abrasive Deburring & Finishing
C071702	2. Abrasive Flow Machining
C071703	3. Abrasive Machining
C071704	4. Grinding
C071705	5. Honing
C071706	6. Superabrasive Machining
C071707	7. Superfinishing
C0718	18. Deburring & Edge Finishing
C071801	1. Abrasive Deburring & Finishing
C071802	2. Lapping
C071803	3. Mass Finishing
C0719	19. Machining Specific Materials
C071901	1. Machining Composites
C071902	2. Machining Ferrous Metals
C071903	3. Machining Nonferrous Metals
C071904	4. Machining Plastic Materials
C0720	20. Metalcutting Fundamentals
C072001	1. Chip Formation Geometry
C072002	2. Machinability & Tool Life
C072003	3. Metalcutting Fundamentals
C0721	21. Nontraditional Machining Processes
C072101	1. Chemical Machining
C072102	2. Electrical Discharge Machining (EDM)
C072103	3. Electrochemical Machining
C0722	22. Tooling
C072201	1. Modular Tooling
C072202	2. Tool Grinding & Sharpening
C07TH	23. Others

CODE	TOPIC NAME
C08	8. Manufacturing Engineering & Management
C0801	1. Capital Investment Planning & Justification
C0802	2. Cost Estimating
C0803	3. Environmental Manufacturing
C0804	4. Industrial Engineering
C0805	5. Inventory Control
C0806	6. Manufacturing Management
C0807	7. Material Handling
C0808	8. Numerical Control Fundamentals
C0809	9. Plant Engineering & Maintenance
C0810	10. Plant Layout
C0811	11. Process Design & Engineering
C0812	12. Production Planning, Scheduling & Control Fundamentals
C0813	13. Tool & Fixture Design
C0814	14. Workplace Safety & Ergonomics
C08TH	15. Others
C09	9. Manufacturing Systems, Automation & IT
C0901	1. CAD/CAM
C0902	2. Internet & E-Manufacturing
C0903	3. Simulation
C090301	1. Optimization
C0904	4. Automation
C090401	1. Automation & Controls
C090402	2. Automation Fundamentals
C090403	3. Flexible Manufacturing Systems (FMS)
C0905	5. System and Process Modeling
C0906	6. Advanced Manufacturing System
C090601	1. Distributed Manufacturing System
C090602	2. Intelligent Manufacturing System
C090603	3. Agile Manufacturing
C090604	4. Virtual Enterprises
C090605	5. Computer Integrated Manufacturing (CIM)
C0907	7. IT Application in Manufacturing
C09TH	8. Others
C10	10. Materials
C1001	1. Ceramics
C1002	2. Composites
C1003	3. Die Materials
C1004	4. Heat Treating Fundamentals
C1005	5. Material Science Fundamentals
C1006	6. Plastic Materials & Compounding
C1007	7. Tribology: Friction, Wear & Lubrication Fundamentals
C1008	8. Wood
C1009	9. Metals

CODE	TOPIC NAME
C100901	1. Aluminum, Aluminum Alloys
C100902	2. Carbides
C100903	3. Copper, Copper Alloys
C100904	4. Nickel, Nickel Alloys
C100905	5. Powder Metallurgy
C100906	6. Stainless Steel
C10TH	10. Others
C11	11. Measurement, Inspection & Testing
C1101	1. Acoustic & Ultrasonic Analysis
C1102	2. Automated Inspection
C1103	3. Geometric Dimensioning & Tolerancing (GD&T)
C1104	4. In-Process Measurement & Inspection
C1105	5. Laser Measurement & Inspection
C1106	6. Materials Testing
C1107	7. Metrology Fundamentals
C1108	8. Non-Destructive Testing (NDT)
C1109	9. Tolerance Analysis
C1110	10. Gaging
C111001	1. Calibration
C111002	2. Dimensional Measurement
C111003	3. Form Measurement
C111004	4. Gage Repeatability & Reliability
C111005	5. Optical Measurement & Inspection
C111006	6. Surface Measurement
C1111	11. Fault Diagnosis
C11TH	12. Others
C12	12. Plastics Molding & Manufacturing
C1201	1. Machining Plastic Materials
C1202	2. Moldmaking
C1203	3. Plastics Assembly
C1204	4. Plastics Finishing
C1205	5. Plastics Molding Fundamentals
C1206	6. Welding Plastic Parts
C1207	7. Molding Processes
C120701	1. Blow Molding
C120702	2. Extrusion
C120703	3. Film Blowing & Casting
C120704	4. Injection Molding
C120705	5. Lamination
C120706	6. Reaction Injection Molding (RIM)
C120707	7. Rotational Molding
C120708	8. Thermoforming
C12TH	8. Others

CODE	TOPIC NAME
C13	13. Product Design Management
C1301	1. Advanced Process & Quality Planning (APQP)
C1302	2. Collaborative Design Technologies
C1303	3. Component Selection & Specification (Purchased Parts)
C1304	4. Design For Automation
C1305	5. Design For Manufacturing & Assembly (DFMA)
C1306	6. Design For Quality
C1307	7. Design For Reliability
C1308	8. Failure Mode & Effects Analysis (FMEA)
C1309	9. Geometric Design
C1310	10. Manufacturing Tolerance & Process Specification
C1311	11. Materials Selection & Specification
C1312	12. Product Lifecycle Management (PLM)
C1313	13. Computer-Aided Design & Engineering
C131301	1. Computer-Aided Design (CAD)
C131302	2. Computer-Aided Engineering (CAE)
C131303	3. Finite Element Analysis (FEA)
C131304	4. Solid Modeling
C13TH	14. Others
C14	14. Quality
C1401	1. Advanced Process & Quality Planning (APQP)
C1402	2. Benchmarking
C1403	3. Continuous Improvement
C1404	4. Process Improvement Techniques
C1405	5. Quality & Inspection of Finishes
C1406	6. Quality Function Deployment (QFD)
C1407	7. Quality Fundamentals
C1408	8. Quality Standards (ISO, QS, AS, Etc)
C1409	9. Total Quality Management (TQM)
C1410	10. Statistical Methods
C141001	1. Design of Experiments (DOE)
C141002	2. Six Sigma
C141003	3. Statistical Process Control
C14TH	11. Others
C15	15. Rapid Prototyping
C1501	1. Deposition Modeling
C1502	2. Laminated Object Modeling (LOM)
C1503	3. Prototyping Fundamentals
C1504	4. Rapid Manufacturing
C1505	5. Rapid Prototyping, Applications
C1506	6. Rapid Tooling
C1507	7. Selective Laser Sintering (SLS)
C1508	8. Stereolithography (SLA)
C15TH	9. Others

CODE	TOPIC NAME
C16	16. Research & Development / New Technologies
C1601	1. Microelectromechanical Systems (MEMS)
C1602	2. Nanotechnology
C1603	3. Research & Development
C1604	4. Technology Transfer
C1605	5. Manufacturing Education
C16TH	6. Others
C17	17. Robotics & Machine Vision
C1701	1. Imaging Technologies
C1702	2. Machine Vision Fundamentals
C1703	3. Robotic Assembly
C1704	4. Robotic Finishing
C1705	5. Robotic Inspection
C1706	6. Robotic Material Handling
C1707	7. Robotic Systems Design
C1708	8. Robotic Welding
C1709	9. Robotics Fundamentals
C17TH	10. Others
C18	18. Welding
C1801	1. Butt Welding
C1802	2. Electron Beam Welding
C1803	3. Friction Welding
C1804	4. Gas Welding
C1805	5. High Energy Beam Welding
C1806	6. Laser Welding
C1807	7. Resistance Welding
C1808	8. Seam Welding
C1809	9. Spot Welding
C1810	10. Ultrasonic Welding
C1811	11. Weld Quality, Testing, & Inspection
C1812	12. Welding Fundamentals
C1813	13. Arc Welding
C181301	1. Arc Welding
C181302	2. Gas Shielded Arc Welding
C181303	3. Plasma Arc Welding
C181304	4. Submerged-Arc Welding
C18TH	14. Others
CTH	19. Others

APPENDIX D – ASSEMBLY OF MCV1 DATASETS

The assembly details of the various subsets of the MCV1 dataset are presented here, to facilitate the verifying of results and subsequent research work with these datasets.

D.1. *abstracts_146*

The assembly of this dataset consists of a few stages.

Step 1: Out of the 1,434 documents, extract documents tagged with only 1 topic label. This will yield 121 documents.

Step 2: Identify the 121 documents by their parent level-1 topic. Out of these documents, extract topics that have at least 9 documents in them. This will yield the following 7 topics:

- C05 – Forming & Fabricating
- C07 – Machining & Material Removal Processes
- C08 – Manufacturing Engineering & Management
- C09 – Manufacturing Systems, Automation & IT
- C10 – Materials
- C16 – Research & Development/New Technologies
- C 17 – Robotics & Machine Vision

At this point, there will be 91 documents distributed across these 7 topics.

Step 3: Return to initial 1,434 documents again, and extract documents tagged with exactly two topics. Out of the returned documents, extract only the documents whose double labels share the same parent level-1 topic. This will yield 79 documents.

Step 4: Identify the returned 79 documents by their parent level-1 topic. Out of these documents, extract only the documents that belong to one of the 7 topics identified in Step 2. This will yield the 55 documents.

Step 5: Combine the 91 documents from Step 2 and 55 documents from Step 4, to form *abstract_146*.

D.2. *abstracts_349*

The documents in this dataset are grouped into 7 level-3 topics:

- C071704 – Grinding
- C072002 – Machinability & Tool Life
- C072003 – Metalcutting Fundamentals
- C090403 – Flexible Manufacturing Systems (FMS)
- C090602 – Intelligent Manufacturing System
- C131301 – Computer-Aided Design (CAD)
- C131303 – Finite Element Analysis (FEA)

From the 1,434 documents in MCV1, we choose all the documents that are tagged with at least 1 of these 7 labels. In this smaller subset of documents, there are documents that are tagged with more than 1 of these 7 labels (since within our coding policy, a document is allowed to have more than 1 topic label). We further filter out these documents, leaving us with a final 349 documents, each having only 1 of these 7 labels.

A document in *abstract_349* may still contain more than 1 topic label, but we only take into account the 7 labels mentioned; in that sense, each document falls into 1 of the 7 topics.

D.3. *abstracts_252*

The documents in this dataset are grouped into 5 level-3 topics:

- C071704 – Grinding
- C072002 – Machinability & Tool Life
- C090403 – Flexible Manufacturing Systems (FMS)
- C090602 – Intelligent Manufacturing System
- C131301 – Computer-Aided Design (CAD)

These topics make up 5 of the 7 topics found in *abstracts_349*. The assembly procedure is similar to that of *abstracts_349*; documents that are tagged with at least 1 of these 5 labels are extracted, and from this smaller subset, we choose documents that have only 1 of these 5 labels. This will give us with 252 documents, spread across these 5 level-3 topics.

The motivation for omitting 2 of the topics (C072003 and C131303) from *abstracts_349* to form *abstracts_252* is to make each topic more distinct and defined from the others. “C072003 – Metalcutting Fundamentals” shares the same level-2 parent topic as “C072002 – Machinability & Tool Life”, and likewise for “C131303 – Finite Element Analysis (FEA)” and “C131301 – Computer-Aided Design (CAD)”. The assembly of this dataset allows us to investigate whether there is an improvement in performance, when the topics within the dataset are conceptually further away from each other.

D.4. *abstracts_160*

The documents in this dataset are grouped into 5 level-2 topics:

- C0715 – Turning
- C0813 – Tool & Fixture Design
- C1004 – Heat Treating Fundamentals
- C1102 – Automated Inspection
- C1505 – Rapid Prototyping, Applications

The assembly is similar to *abstracts_349*, covered in Section C.2. Documents that are tagged with these topics are chosen from the initial 1434 documents in MCV1, and documents that are tagged with more than 1 of these 5 labels are removed from the set. This will leave us with 160 documents, spread across 5 level-2 topics.

D.5. *abstracts_197*

The documents in this dataset are grouped into 5 level-2 topics:

- C0711 – Milling
- C0808 – Numerical Control Fundamentals
- C1007 – Tribology: Friction, Wear & Lubrication Fundamentals
- C1104 – In-process Measurement & Inspection
- C1603 – Research & Development

The assembly is similar to *abstracts_349*, covered in Section C.2. Documents that are tagged with these topics are chosen from the initial 1434 documents in MCV1, and documents that are tagged with more than 1 of these 5 labels are removed from the set. This will leave us with 197 documents, spread across 5 level-2 topics.

D.6. *abstracts_319*

abstracts_319 is made up of documents from the year 1998, in MCV1. The assembly of this dataset is different in that we do not specifically note the topics within the dataset. It is meant to represent a real world dataset of Manufacturing engineering technical papers, with varied and overlapping content. The purpose of assembling this dataset is to evaluate the usefulness of the proposed Topic Detection method on a real world dataset, where prior knowledge of the topic labels is not available.

APPENDIX E – RESULTS FOR *abstracts_319*

This section contains the list of equivalence classes that picked up some of the major distinct topics within the dataset.

Class 1:

- process design
- design manufacturing

Probable topic: Design of manufacturing processes

Class 2:

- systems manufacturing
- control systems

Probable topic: Manufacturing systems

Class 3:

- concurrent engineering
- rapid prototyping

Probable topic: Rapid prototyping

Class 4:

- metal forming
- sheet metal

Probable topic: Sheet metal forming

Class 5:

- product development
- concurrent engineering

Probable topic: Concurrent engineering

Class 6:

- flexible manufacturing fms
- flexible system

Probable topic: Flexible manufacturing system

Class 7:

- tool life
- cutting edge

Probable topic: Tool life