

國立交通大學

資訊科學與工程研究所

博士論文

以模糊理論與高頻項目集為基礎之文件分群研究

Fuzzy Frequent Itemset-based Textual Document Clustering

研究生：陳淳齡

指導教授：梁 婷 博士

曾守正 博士

中華民國九十九年七月

以模糊理論與高頻項目集為基礎之文件分群研究

Fuzzy Frequent Itemset-based Textual Document Clustering

研究生：陳淳齡

Student : Chun-Ling Chen

指導教授：梁 婷 博士

Advisor : Dr. Tyne Liang

曾守正 博士

Dr. Frank S.C. Tseng

國立交通大學

資訊科學與工程研究所



A Dissertation Submitted to
Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

July 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年七月

以模糊理論與高頻項目集為基礎之文件分群研究

學生：陳淳齡

指導教授：梁 婷 博士

曾守正 博士

國立交通大學 資訊科學與工程研究所

摘 要

隨著文字類型文件的數量大幅成長，文件分群技術可用來有效管理這些數量龐大的文件，以便於日後的檢索及瀏覽。為了提升文件分群品質，近年來陸續有學者採用關聯規則探勘技術所產生之高頻項目集於文件分群方法中，解決了一般在文件分群中常遇到的高維度詞彙、執行效能、分群正確性、和自動產生有意義之群集標籤等多項問題。然而，採用關聯規則探勘技術較容易忽略重要且出現頻率較少的關鍵詞彙，再者如項目間的關係程度太高，也會產生數量過多的高頻項目集，造成分群執行時間過長。因此，本研究提出三個以模糊理論和高頻項目集為基礎的文件分群方法，主要是利用模糊關聯規則探勘技術所產生之模糊高頻項目集來有效降低詞彙維度，並可依每個詞彙在文件集中的散佈情況和出現頻率，區分為高頻詞、中頻詞或低頻詞。

本研究首先提出 Fuzzy Frequent Itemset-based Hierarchical Document Clustering (F^2IHC) 方法，主要是利用模糊關聯規則探勘技術找出關鍵詞彙間的關聯性，進而以模糊高頻項目集來產生候選群集，並藉由計算文件與候選群集間的相似度來進行文件分群。此外，並將分群結果以階層式群集樹來呈現，使得歸類好的群集具有容易瀏覽的特性。第二，為了能使用具概念性詞彙來自動標註為群集標籤，我們提出 Fuzzy Frequent Itemset-based Document Clustering (F^2IDC) 方

法，此方法結合 WordNet 探索關鍵詞彙間的語意關係，並加入從 WordNet 中對應出的上位詞 (hypernyms)於文件中，進而擷取出具概念性的群集標籤來表示群集主題。第三，我們提出 Fuzzy Frequent Itemset-based Soft Clustering (F^2ISC) 方法，此方法主要是擴充 F^2IDC 方法，並採用模糊理論之 α -cut 法，能使一份文件分群到一至多個群集中。

在本研究的文件分群過程中，由於使用模糊高頻項目集降低詞彙維度，且所產生之模糊高頻項目集並不會隨著文件數而增加，所以可有效地應用於大文件集的分群上。與傳統的分群方法相比較，實驗結果顯示本論文所提出之研究方法，能有效提高文件分群的正確性與效能，使得文件分群效果更加完善。

關鍵字：文件分群、文字探勘、關聯規則探勘、高頻項目集、模糊集合理論、
WordNet



Fuzzy Frequent Itemset-based Textual Document Clustering

Student: Chun-Ling Chen

Advisors: Dr. Tyne Liang

Dr. Frank S.C. Tseng

Institute of Computer Science and Engineering

National Chiao Tung University

ABSTRACT

With the rapid growth of text documents, document clustering technique is emerging for efficient document retrieval and better document browsing. Recently, some methods had been proposed to resolve the problems of high dimensionality, scalability, accuracy, and meaningful cluster labels by using frequent itemsets derived from association rule mining for clustering documents. However, there are still two situations to be confronted, if we use association rule mining in our approaches: (1) the important sparse key terms may be obscured; (2) too many itemsets will be produced, especially when items in the dataset are highly correlated. Moreover, frequent itemset-based clustering methods usually need a lot of time to generate the large number of itemsets. Considering the above two issues, we present three fuzzy frequent itemset-based document clustering approaches which using fuzzy association rule mining to provide significant dimensionality reduction over interesting fuzzy frequent itemsets. By applying fuzzy association rule mining, each term in the document dataset is labeled with a linguistic term, like *Low*, *Mid*, or *High*.

First, we propose the Fuzzy Frequent Itemset-based Hierarchical Document

Clustering (F²IHC) approach, which employ fuzzy set theory for document representation to find suitable fuzzy frequent itemsets for clustering documents. In addition, F²IHC constructs a hierarchical cluster tree for providing flexible browsing. Second, in order to label clusters with conceptual terms, we present a Fuzzy Frequent Itemset-based Document Clustering (F²IDC) approach with the use of WordNet as background knowledge to explore better ways of representing document semantically for clustering. F²IDC presents a means of dynamically deriving a hierarchical organization of hypernymy from WordNet based on the content of each document without use of training data or standard clustering techniques. Third, we propose a Fuzzy Frequent Itemset-based Soft Clustering (F²ISC) approach by extending F²IDC under the consideration of overlapping clusters. F²ISC provides an accurate measure of confidence, and adopts the α -cut concept to assign each document to one or more than one cluster.

As a result, in the proposed clustering approaches, the interesting fuzzy frequent itemsets are used to reduce the dimensionality of term vectors. In addition, these itemsets do not increase with the growth of documents. Hence, our approaches perform better for large document collections. Our experimental results show that our proposed F²IHC, F²IDC, and F²ISC approaches indeed provide more accurate clustering results than prior influential clustering methods presented in recent literature.

Keywords: *Document Clustering, Text Mining, Association Rule Mining, Frequent Itemsets, Fuzzy Set Theory, WordNet.*

ACKNOWLEDGEMENT

(誌 謝)

隨著論文的付梓，博士班學生生活也即將劃上句點，迎向人生另一個階段。在這段時間裡，除了獲得專業知識外，更令人驚喜地獲得不少歡笑與淚水交織纏繞出溫暖又深刻的回憶，這些回憶都將深刻地烙印於我心深處。論文得以順利完成，都要歸功於這一路上關心、鼓勵和幫助過我的人，在此致上最真誠的感謝。

首先，我最感謝的是指導教授梁婷老師以及共同指導教授曾守正老師，感謝兩位老師孜孜不倦的給予學生諸多意見與指導，在這段期間提供學生一個可以充分表現自我的舞台，並且不斷的給予學生機會，使我成長茁壯。於論文寫作期間，感謝梁婷老師在論文架構和實驗設計等方面給予我精闢的見解與建議；感謝曾守正老師於論文與理論架構、研究方法與英文寫作上提供我細心的指導與幫忙，使得本篇論文能更趨完善。

此外，還要感謝所上的李素瑛教授、胡毓志教授和彭文志教授在各階段口試時，所提供之許多寶貴意見。同時感謝校外口試委員：台灣科技大學資工所陳錫明教授、高雄大學資工所洪宗貝教授、成功大學資工所曾新穆教授和台灣大學資管所魏志平教授在口試過程中所提供之許多寶貴建議，讓本篇論文最後益臻完善。諸位口試委員都是我在學術研究的道路上的最佳學習典範。

資訊擷取實驗室的學長、同學和學弟妹都是我博士班研究生涯的好伙伴，謝謝大家這段時間以來的幫忙與陪伴，也祝福你們在生活上都能平安如意。此外，幸運的我還要感謝一群老朋友，感謝再忙也要陪我聊天幫我趕走沮喪的祥賓；感謝陪我和老公到處旅遊的好朋友兼最佳攝影師豪爺；感謝不時關心我且可隨時陪我分享心情的梅詩和金蓉，謝謝你們在各自忙碌的生活裡，還能從遠方稍來關心，讓我更有力量。當然，不能忘記交大公主幫(菁偉、小龍、淑君、妹

妹珮華與又巧)，謝謝你們在生活上對我的關心，不論是吃喝或是玩耍，讓我充滿歡笑能量，真的愛死你們了!

最重要的要感謝我的家人為我提供一個溫暖的家，謝謝爸爸和媽媽對我的寵愛與栽培；謝謝妹夫又正、妹妹珮華和弟弟品宏對我的支持，不論我遭遇任何挫折你們都在我身邊給予我鼓勵，讓我在求學的過程中可以全心專注在學術研究上，謝謝你們。

最後，要感謝我的老公連進給予我默默的呵護與關心！因為緣分讓我們在2007年7月7號相遇，更因為相互了解讓我們在2010年1月10號能一起牽手成為人生伴侶，也讓我擁有更多愛護我的家人：高壽好客的阿公、體貼的公公婆婆和超有話聊的弟弟連欽，在此與你們共同分享這份喜悅與榮耀。

僅以此論文，獻給我最親愛的家人。



TABLE OF CONTENTS

| | |
|---|-----------|
| 摘要 | i |
| ABSTRACT..... | iii |
| ACKNOWLEDGEMENT..... | v |
| TABLE OF CONTENTS..... | vii |
| LIST OF FIGURES | ix |
| LIST OF TABLES..... | xi |
| LIST OF NOTATIONS..... | xiii |
| Chapter 1 Introduction | 1 |
| 1.1 Background and Motivation..... | 1 |
| 1.2 Research Objectives | 4 |
| 1.3 Organization of the Thesis | 6 |
| Chapter 2 Related Work..... | 7 |
| 2.1 A Generic Process of Document Clustering..... | 7 |
| 2.2 Document Clustering Methods..... | 9 |
| 2.3 Association Rules for Text Mining Applications..... | 12 |
| 2.4 Fuzzy Set Theory..... | 15 |
| Chapter 3 Fuzzy Frequent Itemset-based Hierarchical Document Clustering (F²IHC) Approach..... | 17 |
| 3.1 Stage 1: Document Pre-processing..... | 18 |
| 3.2 Stage 2: Candidate Clusters Extraction..... | 21 |
| 3.2.1 The Membership Functions..... | 22 |
| 3.2.2 The Fuzzy Association Rule Mining Algorithm for Text | 24 |
| 3.2.3 An Illustrative Example of Stage 2 | 26 |
| 3.3 Stage 3: The Cluster Tree Construction | 31 |
| 3.3.1 Building the Document-Cluster Matrix (DCM)..... | 32 |
| 3.3.2 Building the Hierarchical Cluster Tree..... | 34 |
| 3.3.3 Tree Pruning | 35 |
| 3.3.4 An Illustrative Example of Stage 3..... | 37 |
| 3.4 Experiments..... | 42 |
| 3.4.1 Datasets | 42 |
| 3.4.2 Evaluation of Cluster Quality: Overall F-measure..... | 44 |
| 3.4.3 The Effect of Feature Selection..... | 45 |
| 3.4.4 Experimental Results and Analysis..... | 46 |
| 3.5 Summary | 52 |

| | |
|--|-----------|
| Chapter 4 Fuzzy Frequent Itemset-based Document Clustering (F²IDC) Approach..... | 53 |
| 4.1 Stage 1: Document Analyzing..... | 54 |
| 4.2 Stage 2: Document Representation and Enrichment..... | 55 |
| 4.3 Stage 3: Document Clustering..... | 58 |
| 4.3.1 The Fuzzy Association Rule Mining Algorithm for Texts..... | 58 |
| 4.3.2 Clustering | 59 |
| 4.4 An Illustrative Example of F ² IDC Method | 60 |
| 4.5 Experiments..... | 62 |
| 4.5.1 Datasets | 63 |
| 4.5.2 Parameters Selection..... | 64 |
| 4.5.3 Experimental Results and Analysis | 66 |
| 4.6 Summary | 73 |
| Chapter 5 Fuzzy Frequent Itemset-based Soft Clustering (F²ISC) Approach .. | 75 |
| 5.1 Document Analysis Module..... | 76 |
| 5.2 TermOnto Construction Module..... | 77 |
| 5.3 Candidate Cluster Extraction Module | 78 |
| 5.4 Overlapping Cluster Generation Module..... | 78 |
| 5.5 An Illustrative Example of F ² ISC Method..... | 79 |
| 5.6 Experiments..... | 82 |
| 5.6.1 Parameters Selection..... | 83 |
| 5.6.2 Experimental Results and Analysis | 84 |
| 5.7 Summary | 89 |
| Chapter 6 Conclusions and Future Work | 91 |
| 6.1 Conclusions | 91 |
| 6.2 Future Work | 92 |
| Bibliography | 94 |

LIST OF FIGURES

| | |
|--|----|
| Figure 2-1: General process of document clustering. | 7 |
| Figure 2-2: Types of document classification..... | 9 |
| Figure 3-1: The F ² IHC framework. | 17 |
| Figure 3-2: A detailed illustration of Algorithm 3.1..... | 21 |
| Figure 3-3: The predefined membership functions of this example..... | 23 |
| Figure 3-4: A detailed illustration of Algorithm 3-2. | 25 |
| Figure 3-5: A formal illustration of Document-Term Matrix..... | 32 |
| Figure 3-6: A formal illustration of Term-Cluster Matrix..... | 33 |
| Figure 3-7: A formal illustration of Document-Cluster Matrix..... | 34 |
| Figure 3-8: A detailed illustration of Algorithm 3.3..... | 37 |
| Figure 3-9: The derived hierarchical cluster tree..... | 41 |
| Figure 3-10: The accuracy test of F ² IHC for different MinSup values with the optimal cluster numbers determined by the sibling merging algorithm. | 49 |
| Figure 3-11: The detailed time cost analysis of F ² IHC on five datasets..... | 50 |
| Figure 3-12: Scalability of F ² IHC..... | 51 |
| Figure 4-1: The F ² IDC framework. | 54 |
| Figure 4-2: The detailed description of Algorithm 4.1..... | 57 |
| Figure 4-3: The predefined membership functions..... | 59 |
| Figure 4-4: The detailed description of Algorithm 4.2..... | 60 |
| Figure 4-5: The process of Algorithm 4.1 of this example..... | 61 |
| Figure 4-6: The process of Algorithm 3.2 of this example..... | 61 |
| Figure 4-7: The process of Algorithm 4.2 of this example..... | 62 |
| Figure 4-8: The accuracy test of F ² IDC for different MinSup values with the optimal cluster numbers determined by the clusters merging step algorithm. | 72 |
| Figure 4-9: Scalability of F ² IDC..... | 73 |

Figure 5-1: The F²ISC framework. 76

Figure 5-2: A formal illustration of Multiple Clusters Matrix..... 79

Figure 5-3: The detailed description of Algorithm 5.1..... 80

Figure 5-4: The process of Algorithm 4.1 of this example..... 81

Figure 5-5: The process of Algorithm 4.2 of this example..... 81

Figure 5-6: The process of Algorithm 5-1 of this example. 82

Figure 5-8: The detailed time cost analysis of F²ISC on Reuters dataset..... 89



LIST OF TABLES

| | |
|--|----|
| Table 2-1: Summary for our approaches and the other document clustering algorithms. | 12 |
| Table 3-1: Document set. | 21 |
| Table 3-2: The fuzzy set in this example. | 26 |
| Table 3-3: The count values of three fuzzy regions for each key term. | 27 |
| Table 3-4: The set of fuzzy frequent 1-itemsets in this example. | 27 |
| Table 3-5: The candidate set C_2 | 28 |
| Table 3-6: The fuzzy values of (stock.Low, record.Low) in D | 28 |
| Table 3-7: The count values of candidate 2-itemsets. | 29 |
| Table 3-8: The DTM of this example. | 38 |
| Table 3-9: The TCM of this example. | 38 |
| Table 3-10: The DCM of this example. | 39 |
| Table 3-11: The <i>Inter_Sim</i> values of all target clusters. | 39 |
| Table 3-12: The compare results between the parent cluster $c_{(medical)}^1$ | 41 |
| Table 3-13: Statistics for our test datasets. | 44 |
| Table 3-14: Keyword statistics of our test datasets. | 46 |
| Table 3-15: Comparison of the overall F-Measure. | 47 |
| Table 4-1: Statistics for our test datasets. | 64 |
| Table 4-2: List of all parameters for our algorithms and the other three algorithms. ... | 65 |
| Table 4-3: Keyword statistics of our test datasets. | 65 |
| Table 4-4: Average overall F-measure comparison for four clustering algorithms. | 67 |
| Table 4-5: Improvement Ratio for other three clustering algorithms on the four datasets. | 68 |
| Table 4-6: The effect of enriching the document representation. | 70 |
| Table 4-7: Cluster Labels generated by F ² IDC algorithm on Re0 dataset. | 70 |

| | |
|--|----|
| Table 5-1: List of all parameters for our algorithms and the other four algorithms. ... | 83 |
| Table 5-2: Average overall F-measure comparison for five clustering algorithms on the four datasets..... | 85 |
| Table 5-3: The effect of enriching the document representation on <i>Classic</i> and <i>Re0</i> datasets. | 88 |
| Table 5-4: The effect of enriching the document representation on <i>R8</i> and <i>Webkb</i> datasets. | 88 |



LIST OF NOTATIONS

| | |
|--|--|
| D | A document set |
| n | The number of documents |
| d_i | The i -th document, $1 \leq i \leq n$ |
| T | The term set of D |
| m | The number of key terms in D |
| t_j | The j -th key term, $1 \leq j \leq m$ |
| K_D | The key term set of D |
| f_{ij} | The frequency of key tem t_j in document d_i , $1 \leq i \leq n$, $1 \leq j \leq m$, |
| r | The fuzzy region, and $r \in \{Low, Mid, High\}$ |
| $w_{ij}^r(f_{ij})$ | The fuzzy value converted from f_{ij} in region r |
| $count_j^r$ | the summation of w_{ij} values for $i = 1$ to n |
| $max-count_j$ | The maximum count value among $count_j^r$ values |
| $max-R_j$ | The fuzzy region of t_j with $max-count_j$ |
| k | The number of candidate clusters of D |
| τ | The fuzzy frequent itemsets for describing \tilde{c} |
| $\tilde{c}_{(\tau)}^q = (\tilde{D}_c, \tau)$ | A candidate cluster, with key term set $\tau = \{t_1, t_2, \dots, t_q\} \subseteq K_D$ |
| \tilde{C}_D | The candidate cluster set of D |
| c_i^q | A target cluster, with key term set $\tau = \{t_1, t_2, \dots, t_q\} \subseteq K_D$ |
| C_D | The target cluster set of D |
| CT | The cluster tree of D |
| \mathcal{F} | A term forest of a set of terms $\{t_1, t_2, \dots, t_i, \dots, t_m\} \subseteq K_D$ |
| \mathcal{J} | A term tree of term t_j |
| $W = [w_{ij}^{\max-R_j}]$ | The Document-Term Matrix (DTM) |
| $G = [g_{jl}^{\max-R_j}]$ | The Term-Cluster Matrix (TCM) |
| $V = [v_{il}]$ | The Document-Cluster Matrix (DCM) |
| $M = [m_{ig}]$ | The Multiple Clusters Matrix (MCM) |
| $Inter_Sim(c_x^1, c_y^1)$ | The inter-cluster similarity between two target clusters c_x^1 and c_y^1 |

Chapter 1

Introduction

1.1 Background and Motivation

Clustering textual documents into different groups is an important step in indexing, retrieval, management, and mining of abundant text data on the Web or in corporate document management repositories [4][27][56][61]. Recently, the incessant flourishing of Internet invigorates various textual documents to be shared over the cyberspace astonishingly. However, it also makes users suffer from the information-overloading problem. In particular, when users pose queries to WWW search engines, they usually bewilderingly receive a small number of relevant Web pages intermingled with a large number of irrelevant Web pages. The focus of textual document clustering technique has shifted towards providing ways to reorganize search results into meaningful cluster hierarchies for efficiently browse large collections of documents. Therefore, a good textual document clustering technique has to provide a helpful complement for traditional search engines when keyword-based search returns too many documents.

The aim of document clustering algorithms is to automatically discover the hidden similarity and the key concepts of clustered documents for users to comprehend a large amount of documents. Over the past decades, several effective document clustering algorithms have been proposed to mitigate the hassle, including the k -means [36], Bisecting k -means [53], Hierarchical Agglomerative Clustering (HAC) [26][29][61], and Unweighted Pair Group Method with Arithmetic Mean

(UPGMA) [39]. Nevertheless, as pointed out by [3][17][24][45][33], there are still challenges in improving the clustering quality, which we list as follows:

- (1) *To cope with high dimensionality*: As the volume of textual document increases, the dimensionality of term features increases as well.
- (2) *To improve the scalability*: Many document clustering algorithms work fine on small document sets, but fail to deal with large document sets efficiently.
- (3) *To promote the accuracy*: Many existing document clustering algorithms require users to specify the number of clusters as an input parameter. However, it is difficult to determine the number of clusters in advance. Moreover, an incorrect estimation of the input parameter, i.e., the number of clusters, may lead to poor clustering accuracy [17].
- (4) *To assign meaningful cluster labels*: Meaningful cluster labels will guide users in the process of browsing the retrieved results. Thus, each cluster should be labeled with an understandable description. However, most of traditional clustering algorithms do not provide labels for clusters.
- (5) *To extract semantics from text*: The bag-of-words representation used for clustering algorithms is often unsatisfactory as it ignores the conceptual similarity of terms that do not co-occur actually [24][45].
- (6) *To enable overlapping clusters*: Many well-known clustering algorithms focus on hard clustering, where each document belongs to exactly one cluster. However, a document could contain multiple subjects. By using soft clustering algorithms [33], a document would appear in multiple clusters (i.e., overlapping clusters).

To resolve the problems of high dimensionality, large size, and understandable cluster description, Beil *et al.* [3] developed the first frequent itemset-based algorithm, namely Hierarchical Frequent Term-based Clustering (HFTC), where the frequent

itemsets are generated based on the association rule mining [12]. They only considered the low-dimensional frequent itemsets as clusters. Moreover, HFTC discovers overlapping clusters, which is useful for a search engine where overlapping clusters occur like Yahoo! Directory.

However, the experiments of Fung *et al.* [17] showed that HFTC is not scalable. For a scalable algorithm, Fung *et al.* proposed the FIHC (Frequent Itemset-based Hierarchical Clustering) algorithm by using frequent itemsets derived from association rule mining to construct a hierarchical topic tree for clusters. They also proved that using frequent itemsets for document clustering can reduce the dimensionality of term vectors effectively. Yu *et al.* [63] presented another frequent itemset-based algorithm, called TDC, to improve the clustering quality and scalability. This algorithm dynamically generates a topic directory from a document set using only closed frequent itemsets and further reduces dimensionality. But, the clusters generated by FIHC and TDC are non-overlapping. In [23], the authors proposed that document clustering methods should provide multiple subjective perspectives onto the same document to enhance their practical applicability.

Recently, WordNet [40], one of the most widely adopted thesaurus for English, has been extensively used as an ontology in grouping documents with its semantic relations of terms [24][45][11][28]. Many existing document clustering algorithms mainly transform textual documents into simplistic flat bags of document representation, i.e., term vectors or bag-of-words. Once terms are treated as individual items in such simplistic representation, the semantic content of a document is decomposed and cannot be reflected. Thus, Dave *et al.* [11] proposed using synsets as features for document representation and subsequent clustering. However, synsets decrease the clustering performance in all experiments without considering word

sense disambiguation. Meanwhile, Hotho *et al.* [24] used WordNet in document clustering for word sense disambiguation to improve the clustering results. Jing *et al.* [28] presented another application of WordNet, which described how to find mutual information between terms by using the background knowledge through WordNet. In [45], Recuperó proposed a new unsupervised document clustering method by using WordNet lexical and conceptual relations to allow common clustering algorithms to perform well. In this thesis, the reasons of utilizing hypernyms from WordNet are two-fold:

- (1) We intend to obtain more general and conceptual labels for derived clusters.
- (2) From the experimental results in [11][49], the authors found that the performance of adding hypernyms is better than adding synonymy.

1.2 Research Objectives

Among the techniques developed for data and text mining, association rule mining [1][20] is one of the useful and successful techniques for discovering interesting rules. It helps users discover meaningful association rules to represent a relationship between different pairs of a set of attribute values. However, there are still two situations to be confronted, if we use association rule mining in our approach:

- (1) Some important terms that express the topics of a document may be rarely appeared in the document collection. That is, only the terms which frequently occur in the document collection can be obtained, which implies the important sparse terms may be obscured in the process of document clustering.

(2) Association rule mining often suffers from producing too many itemsets, especially when items in the dataset are highly correlated [35]. As our approach aims to consider the semantic relationships from WordNet, the situation may become severer after adding correlated hypernyms.

Considering the above two issues, we will propose an approach which stems from prior studies [22][30][38], by integrating fuzzy set concept [64] and association rule mining to provide significant dimensionality reduction over interesting frequent itemsets. Moreover, Kaya *et al.* [30] think that fuzzy association rule mining is understandable to humans because it integrates linguistic terms with fuzzy sets. By applying fuzzy association rule mining, we can discover fuzzy frequent itemsets as candidate clusters, like $(term_1.Low, term_2.High)$ or $(term_1.Low, term_2.Low)$, and label the terms with a linguistic term, like *Low*, *Mid*, or *High*.

Thus, we present three document clustering approaches based on fuzzy frequent itemsets. First, we propose the Fuzzy Frequent Itemset-based Hierarchical Document Clustering (F²IHC) approach to solve high dimensionality, scalability, accuracy, and meaningful cluster labels. In addition, F²IHC provides a term-based algorithm for the analysis of a document set to generate a flexible hierarchical document cluster tree, which can be easily integrated into a document management system for providing flexible browsing and retrieving of various applications.

Second, in order to label clusters with conceptual terms, we present a Fuzzy Frequent Itemset-based Document Clustering (F²IDC) approach with the use of WordNet as background knowledge to explore better ways of representing document semantically for clustering. F²IDC presents a means of dynamically deriving a hierarchical organization of hypernymy from WordNet based on the content of each document without use of training data or standard clustering techniques.

Third, we present a Fuzzy Frequent Itemset-based Soft Clustering (F^2ISC) approach by extending F^2IDC under the consideration of overlapping clusters. F^2ISC provides an accurate measure of confidence, and adopts the α -cut concept [64] to assign each document to one or more than one cluster.

By conducting experimental evaluations on the several datasets, it has been proven that our proposed F^2IHC , F^2IDC , and F^2ISC approaches indeed provide a more accurate cluster result than previous clustering methods presented in recent literature.

1.3 Organization of the Thesis

The subsequent sections of this thesis are organized as follows. In Chapter 2, we briefly review related work on general process of document clustering, major document clustering methods, association rules for text mining Applications, and fuzzy set theory. In Chapter 3, the Fuzzy Frequent Itemset-based Hierarchical Document Clustering (F^2IHC) approach will be described, together with an illustrative example. Chapter 4 illustrates the Fuzzy Frequent Itemset-based Document Clustering (F^2IDC) approach. We depict in Chapter 5 the description of the Fuzzy Frequent Itemset-based Soft Clustering (F^2ISC) approach. Finally, we conclude and propose some future directions in Chapter 6.

Chapter 2

Related Work

In the first place, the general process of document clustering is described in Section 2.1. Then, the literature concerning document clustering methods will be surveyed in Section 2.2. In Section 2.3, we will discuss how association rules are applied to text mining. Finally, we briefly review some basic knowledge of fuzzy sets in Section 2.4

2.1 A Generic Process of Document Clustering

The aim of document clustering is to group similar documents together based on the content of a set of documents. According to [59], we divide the general process of document clustering into three main stages, including *Document Pre-processing*, *Document Representation*, and *Document Clustering* (as shown in Figure 2-1). These stages are described as follows.

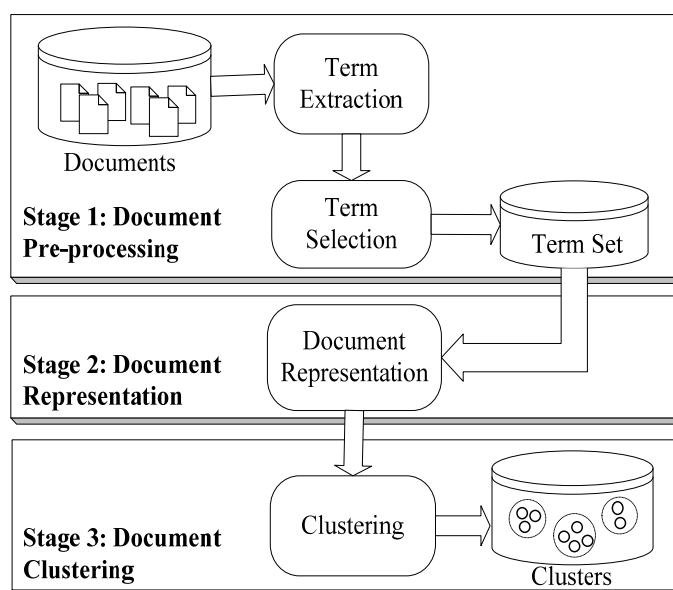


Figure 2-1: General process of document clustering.

1. *Document Pre-processing*. In order to satisfy document clustering methods, the given unstructured documents need to be preprocessed. There are two steps in this stage, namely *Term Extraction* and *Term Selection*, for generating the term set from the document collection.

(1) *Term Extraction*: The whole extraction process is as follows:

- *Extract terms*. Divide the sentences into terms and extract terms as features.
- *Remove the stop words*. A pre-defined stop-word list¹ is applied to remove commonly used words that do not discriminate for topics.
- *Conduct word stemming*. Use the developed stemming algorithms, such as Porter [44], to convert a word to its stem or root form. The frequencies of stemmed terms instead of the original terms in the document collection are computed.

(2) *Term Selection*: After extracting terms, it is crucial to reduce the set of term features, a process referred to as term selection. For example, a term should be discarded (i.e. from the term set) if it appears rarely or more frequently in the document collection. Several methods, such as itemset pruning [3], feature clustering or co-clustering [37], feature selection technique [51], and matrix factorization [50][62], have been applied to reduce the dimensionality for high clustering accuracy.

2. *Document Representation*. The most common representation is the so-called “bag-of-words” matrix, where each document is represented as a vector based on the terms which occur in the relative documents, and then the clustering methods compute the similarity between the vectors [47]. Several document representation

¹ It contains a list of 571 stop words that was developed by the SMART project.

methods have been proposed, including binary (which shows the presence or absence of a term in a document) and term frequency (which shows the frequency of a term in a document).

3. *Document Clustering*. Common approaches for document clustering have been used, including the *k*-means [36], Bisecting *k*-means [53], Hierarchical Agglomerative Clustering (HAC) [26][29][61], and Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [39], etc. The details of each clustering approach will be depicted in the following section.

2.2 Document Clustering Methods

The basic principle of document classification is to classify or group a set of unlabeled documents into classes or clusters. According to [53], we divide document classification into three subcategories, i.e., supervised or unsupervised, hard or soft, and partitioning, hierarchical, or frequent itemset-based. These subcategories can be shown in a tree structure as Figure 2-2 depicts, which we describe as follows.

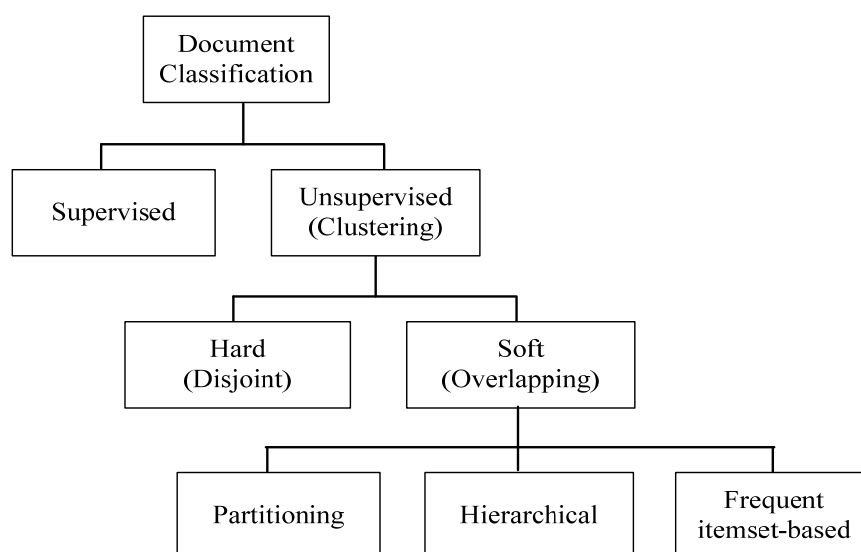


Figure 2-2: Types of document classification.

- 1 **Supervised and Unsupervised (Clustering):** In supervised document classification, a set of predefined classes are available. On the other hand, in unsupervised document classification, also called document clustering, there are no pre-determined classes available. Document clustering is the process of calculating document similarities to form clusters. The documents within a cluster are similar to each other and, simultaneously, dissimilar to the documents in the other groups.
- 2 **Hard (Disjoint) and Soft (Overlapping):** Hard clustering algorithms compute the hard assignment (i.e., each document is assigned to exactly one cluster) and produce a set of disjoint clusters. Soft clustering algorithms compute the soft assignment (i.e., each document allows to appear in multiple clusters) and generate a set of overlapping clusters. For instance, a document discussing “Natural language and Information Retrieval” should be assigned to both of the clusters “Natural language” and “Information Retrieval”.
- 3 **Partitioning, Hierarchical, and Frequent itemset-based:** For document clustering, partitioning-based methods exclusively partition the set of documents into a number of clusters by moving documents from one cluster to another, such as k -means [36] and Bisecting k -means [53].

Compacted to partitioning-based methods, hierarchical-based document clustering is to build a hierarchical tree of clusters whose leaf nodes represent the subset of a document collection, like Hierarchical Agglomerative Clustering (HAC) [26][29][61] and Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [39]. Moreover, this method can be further classified into agglomerative and divisive approaches, which work in a bottom-up and top-down fashion, respectively. An agglomerative clustering iteratively merges two most

similar clusters until a terminative condition is satisfied. On the other hand, a divisive method starts with one cluster, which consists of all documents, and recursively splits one cluster into smaller sub-clusters until some termination criterion is fulfilled.

Besides, a new category of document clustering, namely “frequent itemset-based clustering,” has been extensively developed, including FIHC [17], HFTC [3], and TDC [63]. Frequent itemset-based clustering methods use frequent itemsets generated by the association rule mining and further cluster the documents according to these extracted frequent itemsets. These methods reduce the dimensionality of term features efficiently for very large datasets, thus they can improve the accuracy and scalability of the clustering algorithms. The organization of clusters generated by frequent itemset-based clustering methods could be a flat set or a hierarchical tree of clusters.

Moreover, an advantage of frequent itemset-based clustering method is that each cluster can be labeled by the obtained frequent itemsets shared by the documents in the same cluster. A cluster label could only be used to describe the main concept of the cluster, but also differentiate the cluster from its sibling and parent clusters [55][65]. However, most frequent itemset-based clustering methods ignore the semantics of the terms in the process of generating frequent itemsets. In the thesis, the proposed approaches provides more general cluster labels because they take into account the semantics of the terms using background knowledge, WordNet.

Table 2-1 summarizes the characteristics of the proposed approaches and other document clustering algorithms.

Table 2-1: Summary for our approaches and the other document clustering algorithms.

| | Hierarchical-based | Partitioning-based | Frequent itemset-based |
|-------------|--|--|---|
| Hard | <ul style="list-style-type: none"> • Hierarchical Agglomerative Clustering (HAC) [61] • Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [39] | <ul style="list-style-type: none"> • k-means [36] • Bisecting k-means [53] • Hotho <i>et al.</i> (2003) [24] ★ • Sedding <i>et al.</i> (2004) [49] ★ • Wang <i>et al.</i> (2006) [58] ★ • Recupero (2007) [45] ★ | <ul style="list-style-type: none"> <u>A Hierarchical Tree of Clusters</u> • Fung <i>et al.</i> (2003) [17] • The proposed approach (F²IHC) [8][9] <u>A Flat Set of Clusters</u> • Yu <i>et al.</i> (2004) [63] • The proposed approach (F²IDC) [5][6] ★ |
| Soft | | <ul style="list-style-type: none"> • Lin and Kondadadi (2001) [33] | <ul style="list-style-type: none"> <u>A Hierarchical Tree of Clusters</u> • Beil <i>et al.</i> (2002) [3] <u>A Flat Set of Clusters</u> • The proposed approach (F²ISC) [7] ★ |

★ means a WordNet-based document clustering approach.

2.3 Association Rules for Text Mining Applications

According to [15], the authors have defined that knowledge discovery in database has several interactive and iterative phases to extract useful knowledge from huge volumes of data, where data mining has been recognized as the most important phase, as it offers flexibility for extracting useful patterns from business data.

In data mining, association rule mining [20] is a popular method for discovering interesting association rules in large databases. The form of an association rule can be represented as $X \rightarrow Y$, where X and Y are sets of items and $X \cap Y = \emptyset$, and is usually adopted for market basket analysis to describe the following meaning: customers that buy product X also buy product Y for satisfying some predefined *minimum support value* and *minimum confidence value*. In general, each itemset has an associated measure of statistical significance called *Support* value, which is the fraction of all

transactions that contain the itemset. For example, an itemset X with support value, $\text{supp}(X) = 0.5$, regards there are 50% of transactions in the dataset containing X . An itemset can be chosen as a *frequent itemset* if its support value is larger than or equal to the predefined *minimum support value*. The *confidence* value of an association rule, denoted $\text{conf}(X \rightarrow Y) = \text{supp}(X \cup Y) / \text{supp}(X)$, is to measure how often items in Y appear in transactions which also contain X . Finally, a rule $X \rightarrow Y$ will be discovered whether its confidence value is larger than or equal to the predefined *minimum confidence value* or not.

Due to the strong need for analyzing the vast amount of textual documents spread over the Internet, text mining is also growing rapidly. By the definition described in [15][52][60], Text Mining, also known as Intelligent Text Analysis, Text Data Mining or Knowledge Discovery in Text (KDT), refers generally to the process of extracting interesting and non-trivial information and knowledge from unstructured text. The main purpose of text mining is to acquire fruitful knowledge from a large document set. It draws on techniques from data mining, computational linguistics, database systems, information retrieval, and artificial intelligence to achieve the goal.

As text mining is much more complex than data mining because text data are inherently unstructured and fuzzy [54], some studies [13][15][34] applied the technique of association rule mining in document management. For example, Feldman and Dagan [15] have presented a Knowledge Discovery in Text (KDT) system, which used the simplest information extraction approach to get interesting information and knowledge from unstructured text collections. Lin *et al.* [34] proposed a method, namely Mining Term Association, to acquire the semantic relations between terms when applying to documents. Moreover, Delgado *et al.* [13] think that association rule mining is the first data mining technique employed in

mining text collections. It is very interesting since many applications related to text processing involve associations and co-occurrence between terms. These works mainly focused on analyzing the co-occurrence terms for document management.

Recently, to flexibly conduct the association rule mining for more applications, some research works [22][30][38] have been proposed to integrate fuzzy set theory [64] and association rule mining for handling items with quantitative values while discovering fuzzy association rules from given transactions. Basically, a fuzzy association rule mining approach proposed by Hong *et al.* [22] first use membership functions to convert quantitative values into a fuzzy set in linguistic terms. Then, the scalar cardinality of each linguistic term on all transactions is calculated. The mining process based on fuzzy counts was used to find interesting association rules. In addition, Hong *et al.* [21] described some fuzzy mining concepts and techniques related to association rules discovery in details, including mining fuzzy association rules, mining fuzzy generalized association rules, and mining both membership function and fuzzy association rules.

In the association rule mining technique, each document merely contains binary terms, meaning that a term either appears in a document or not. However, terms in the documents may be presented with quantitative types, such as term frequency or term weight. In this thesis, we thus focus on employing fuzzy association rule mining devised by Hong *et al.* [22] by regarding a document as a transaction, and those term frequency values in a document as the quantitative values (i.e., the number of purchased items in a transaction) to find the association relationships between terms. To illustrate the usefulness of fuzzy data mining in document clustering, we use fuzzy set concepts to model the term frequency describing the important degree of a term in a document. In contrast with using the crisp set concept, in which a term is either a

member of a document or not, fuzzy set concepts make it possible that a term belongs to a document to a certain degree.

2.4 Fuzzy Set Theory

In this section, we briefly review some basic knowledge of fuzzy sets [64]. According to [68], a fuzzy set is considered as a class with fuzzy boundaries.

Definition 2.1 (Fuzz set): A *fuzzy set* A in the universe of discourse $U = \{u_1, u_2, \dots, u_n\}$ is defined by the membership function μ_A , denoted as $\mu_A(u)$, where $u \in U$. Each element u of U has a membership value, in the closed interval $[0,1]$, given by μ .

$$A = \{u_i, \mu_A(u_i) \mid u_i \in U\}. \quad (2.1)$$

Definition 2.2 (Fuzzy Relation): A *fuzzy relation* R between variables v and w , whose domains are V and W , respectively, is defined by function that map an ordered pair (v, w) in $V \times W$ to its degree in the relation, where is a value between 0 and 1.

$$R = V \times W \rightarrow [0, 1]. \quad (2.2)$$

Let μ_A and μ_B be the membership functions of the fuzzy sets A and B , respectively. In the following, we summarized some fuzzy operations used in this thesis.

Definition 2.3 (Fuzzy Set Union): The union of the fuzzy sets A and B is denoted as $A \cup B$ and is defined by

$$A \cup B = \{(u_i, \mu_{A \cup B}(u_i) \mid \mu_{A \cup B}(u_i) = \text{Max}(\mu_A(u_i), \mu_B(u_i)), u_i \in U\}. \quad (2.3)$$

Definition 2.4 (Fuzzy Set Intersection): The intersection of the fuzzy sets A and B is denoted as $A \cap B$ and is defined by

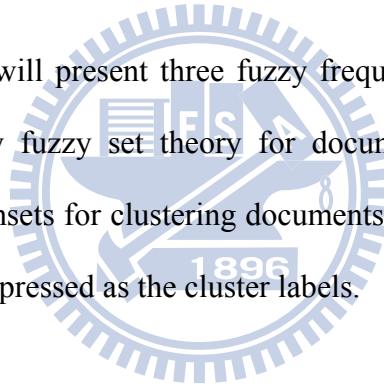
$$A \cap B = \{(u_i, \mu_{A \cap B}(u_i) \mid \mu_{A \cap B}(u_i) = \text{Min}(\mu_A(u_i), \mu_B(u_i)), u_i \in U\} \quad (2.4)$$

Definition 2.5 (α -cut): The α -cut of the fuzzy set A is denoted as A_α and is defined by

$$A_\alpha = \{u_i \mid \mu_A(u_i) \geq \alpha, u_i \in U\} \quad \alpha \in [0,1]. \quad (2.5)$$

The α -cut is the crisp set that contains all the elements of U whose membership values given by μ_A are greater than or equal to the specified value of α .

In the following, we will present three fuzzy frequent itemset-based clustering approaches, which employ fuzzy set theory for document representation, to find suitable fuzzy frequent itemsets for clustering documents. Moreover, the mined fuzzy frequent itemsets will be expressed as the cluster labels.



Chapter 3

Fuzzy Frequent Itemset-based Hierarchical Document Clustering (F²IHC) Approach

In order to browse and organize documents smoothly, hierarchical clustering techniques have been proposed to cluster a collection of documents into a hierarchical tree structure. Despite that, there still exist several challenges for hierarchical document clustering, such as high dimensionality, scalability, accuracy, and meaningful cluster labels [3][16][17].

In this chapter, we will present an effective Fuzzy Frequent Itemset-Based Hierarchical Clustering (F²IHC) approach, which uses fuzzy association rule mining algorithm to construct a hierarchical cluster tree for providing flexible browsing. There are three stages in our F²IHC framework as shown in Figure 3-1. We explain them in Sections 3.1 - 3.3.

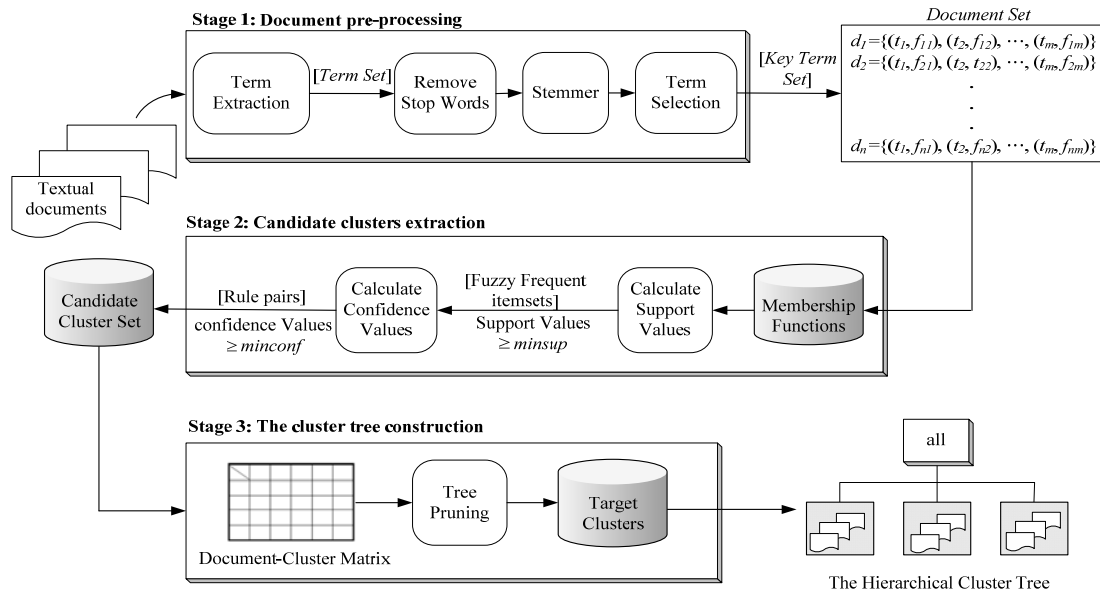


Figure 3-1: The F²IHC framework.

3.1 Stage 1: Document Pre-processing

This stage describes the required transformation processes of documents to obtain the desired representation of documents. As there are thousands of words in a document set, the purpose of this stage is to reduce dimensionality for high clustering accuracy. Several methods, such as itemset pruning [3], feature clustering or co-clustering [37], feature selection technique [51], and matrix factorization [50][62], have been applied to reduce dimensionality. To solve this problem, we have to find the terms that are significant and important to represent the content of each document. Hence, we must remove the terms that are not meaningful and discriminative to increase the clustering accuracy and maintain the computing cost small. We describe the details of the pre-processing in the following:

1. *Divide the sentences into terms.*
2. *Remove the stop words.* We use a stop word list² that contains words to be excluded. The list is applied to remove the terms that have general meaning but do not discriminate for topics.
3. *Conduct word stemming:* Use the developed stemming algorithms, such as Porter [44], to reduce a word to its stem or root form.
4. *Term selection.* The terms with selection metric weights all higher than pre-specified thresholds will be selected as key terms. In our approach, three feature selection methods [46], tf-idf, tf-df, and tfidf-tfdf, are used to select representative terms for each document, and these feature selection methods are defined as follows:

² It contains a list of 571 stop words that was developed by the SMART project.

- (1) tf-idf (term frequency-inverse document frequency): It is denoted as $tfidf_{ij}$ and used for the measure of the importance of term t_j within document d_i . For preventing a bias for longer documents, the weighted frequency of each term is usually normalized by the total frequencies of all terms in document d_i , and is defined as follows:

$$tfidf_{ij} = \frac{f_{ij}}{\sum_{j=1}^m f_{ij}} \times \log\left(\frac{|D|}{|\{d_i | t_j \in d_i, d_i \in D\}|}\right) \quad (3.1)$$

where f_{ij} is the frequency of term t_j in document d_i , and the denominator is the total frequencies of all terms in document d_i . $|D|$ is the total number of documents in the document set D , and $|\{d_i | t_j \in d_i, d_i \in D\}|$ is the number of documents containing term t_j .

- (2) tf-df (term frequency-document frequency): It is represented by $tfdf_{ij}$ and evaluated by (3.2) for the value calculated by dividing the term frequency (TF) by the document frequency (DF), where TF is the number of times a term t_j appears in a document d_i divided by the total frequencies of all terms in d_i , and DF is used to determine the number of documents containing term t_j divided by the total number of documents in the document set D :

$$tfdf_{ij} = TF/DF, \text{ where } TF = \frac{f_{ij}}{\sum_{j=1}^m f_{ij}}, \text{ and } DF = \frac{|\{d_i | t_j \in d_i, d_i \in D\}|}{|D|} \quad (3.2)$$

- (3) tfidf-tfdf: It is the multiplication of $tfidf_{ij}$ and $tfdf_{ij}$, and we denote it as $tfidf-tfdf_{ij}$:

$$tfidf-tfdf_{ij} = tfidf_{ij} * tfdf_{ij} \quad (3.3)$$

After these weights of each term in each document have been calculated, those which have weights all higher than pre-specified thresholds are retained. Subsequently,

these retained terms form a set of key terms for the document set D , and we formally define them as follows.

Definition 3.1: A *document*, denoted $d_i = \{(t_1, f_{i1}), (t_2, f_{i2}), \dots, (t_j, f_{ij}), \dots, (t_m, f_{im})\}$, is a logical unit of text, characterized by a set of key terms t_j together with their corresponding frequency f_{ij} .

Definition 3.2: A *document set*, denoted $D = \{d_1, d_2, \dots, d_i, \dots, d_n\}$, also called a *document collection*, is a set of documents, where n is the total number of documents in D .

Definition 3.3: The *term set* of a document set $D = \{d_1, d_2, \dots, d_i, \dots, d_n\}$, denoted $T_D = \{t_1, t_2, \dots, t_j, \dots, t_s\}$, is the set of terms appeared in D , where s is the total number of terms.

Definition 3.4: The *key term set* of a document set $D = \{d_1, d_2, \dots, d_i, \dots, d_n\}$, denoted $K_D = \{t_1, t_2, \dots, t_j, \dots, t_m\}$, is a subset of the term set T_D , including only meaningful key terms, which are not appeared in a well-defined stop word list, and satisfy the pre-defined minimum threshold of term selection methods.

Based on these definitions, the representation of a document can be derived by Algorithm 3.1 shown in Figure 3-2. For example, for a document set $D = \{d_1, d_2, \dots, d_{10}\}$, which includes ten documents. By Algorithm 3.1, suppose we can obtain the derived representation of D and its key term set $K_D = \{\text{stock, record, profit, medical, treatment, health}\}$ as shown in Table 3-1. Notice that we use a tabular representation, where each entry denotes the frequency of a key term (the column heading) in a document d_i (the row heading), to make our presentation more concise. This representation scheme will be employed in the following to illustrate our approach.

Algorithm 3.1: The document pre-processing algorithm for deriving the representation of all documents in a document set D .

Input: A document set $D = \{d_1, d_2, \dots, d_i, \dots, d_n\}$; A well-defined stop word list; The minimum tf-idf threshold α ; The minimum tf-df threshold β ; The minimum tfidf-tfidf threshold γ ;

Output: The key term set of D , K_D ; The formal representation of all documents in D as defined in Definition 3.1.

1. Extract the term set $T_D = \{t_1, t_2, \dots, t_j, \dots, t_s\}$.
 2. Remove all stop words from T_D .
 3. Apply Word Stemming for T_D .
 4. For each $d_i \in D$ do
 - For each $t_j \in T_D$ do
 - (1) Evaluate its $tfidf_{ij}$, $tfdf_{ij}$, and $tfidf-tfdf_{ij}$ weights.
 - (2) Retain the term if $tfidf_{ij} \geq \alpha$, $tfdf_{ij} \geq \beta$, and $tfidf-tfdf_{ij} \geq \gamma$.
 5. Obtain the key term set K_D based on the previous steps.
 6. For each $d_i \in D$ do
 - For each $t_j \in K_D$ do
 - (1) count its frequency in d_i to obtain $d_i = \{(t_1, f_{i1}), (t_2, f_{i2}), \dots, (t_j, f_{ij}), \dots, (t_m, f_{im})\}$.
-

Figure 3-2: A detailed illustration of Algorithm 3.1.

Table 3-1 : Document set.

| Docs ID | Key Term Set | | | | | |
|----------|--------------|--------|--------|---------|-----------|--------|
| | stock | record | profit | medical | treatment | health |
| d_1 | 2 | 1 | 1 | 0 | 0 | 0 |
| d_2 | 1 | 1 | 0 | 0 | 0 | 0 |
| d_3 | 1 | 0 | 2 | 0 | 0 | 0 |
| d_4 | 0 | 0 | 0 | 3 | 0 | 2 |
| d_5 | 0 | 0 | 0 | 11 | 1 | 1 |
| d_6 | 0 | 1 | 0 | 4 | 0 | 0 |
| d_7 | 0 | 0 | 0 | 8 | 1 | 2 |
| d_8 | 3 | 0 | 1 | 0 | 0 | 0 |
| d_9 | 0 | 1 | 0 | 3 | 0 | 0 |
| d_{10} | 0 | 0 | 0 | 8 | 2 | 1 |

3.2 Stage 2: Candidate Clusters Extraction

The objective of this stage is to take a document set D , a set of predefined membership functions, the minimum support value θ , and the minimum confidence value λ as input, and to output a set of candidate clusters. To achieve this goal, we modified the algorithm proposed by Hong *et al.* [22] to capture the relationships among different key terms of the document set. Since each discovered fuzzy frequent itemset

has an associated fuzzy count value, it can be regarded as the degree of importance that the itemset contributes to the document set.

In the following, we will define the membership functions, present our algorithm, and finally explain our approach by an illustrative example.

3.2.1 The Membership Functions

The membership functions are used to convert each term frequency into a fuzzy set. Therefore, we define the t - f (term frequency) fuzzy set in Definition 3.5 used in this thesis.

Definition 3.5: A t - f fuzzy set of document d_i is a pair (F_{ij}, w_{ij}^r) , where F_{ij} is a set and equals to $\{w_{ij}^{Low}(f_{ij}) / t_j.Low, w_{ij}^{Mid}(f_{ij}) / t_j.Mid, w_{ij}^{High}(f_{ij}) / t_j.High\}$, $w_{ij}^r : F \rightarrow [0, 2]$, and r can be *Low*, *Mid*, or *High*. The notation $t_{j,r}$ is called a fuzzy region of t_j . For each term pair (t_j, f_{ij}) of document d_i , $w_{ij}^r(f_{ij})$ is the grade of membership of t_j in d_i with *Low*, *Mid*, and *High* membership functions.

$$w_{ij}^{Low}(f_{ij}) = \begin{cases} 0, & f_{ij} = 0 \\ 1 + \frac{f_{ij}}{a_1}, & 0 < f_{ij} < a_1 \\ 2, & f_{ij} = a_1 \\ 1 + \frac{a_2 - f_{ij}}{a_2 - a_1}, & a_1 < f_{ij} < a_2 \\ 1, & f_{ij} \geq a_2 \end{cases}, \quad a_1 = \min(f_{ij}), \quad a_2 = \text{avg}(f_{ij}) \quad (3.4)$$

$$w_{ij}^{Mid}(f_{ij}) = \begin{cases} 0, & f_{ij} = 0 \\ 1, & f_{ij} = a_1 \\ 1 + \frac{f_{ij} - a_1}{a_2 - a_1}, & a_1 < f_{ij} < a_2 \\ 2, & f_{ij} = a_2 \\ 1 + \frac{a_3 - f_{ij}}{a_3 - a_2}, & a_2 < f_{ij} < a_3 \\ 1, & f_{ij} = a_3 \end{cases}, \quad a_1 = \min(f_{ij}), a_2 = \text{avg}(f_{ij}), a_3 = \max(f_{ij}) \quad (3.5)$$

$$w_{ij}^{High}(f_{ij}) = \begin{cases} 0, & f_{ij} = 0 \\ 1, & f_{ij} \leq a_1 \\ 1 + \frac{f_{ij} - a_1}{a_2 - a_1}, & a_1 < f_{ij} < a_2 \\ 2, & f_{ij} = a_2 \end{cases}, \quad a_1 = \text{avg}(f_{ij}), a_2 = \max(f_{ij}) \quad (3.6)$$

In formulas (3.4), (3.5), and (3.6), $\min(f_{ij})$ is the minimum frequency of terms in D , $\max(f_{ij})$ is the maximum frequency of terms in D , and $\text{avg}(f_{ij}) = \lceil \frac{\sum_{i=1}^n f_{ij}}{|K|} \rceil$, where $f_{ij} \neq \min(f_{ij})$ or $\max(f_{ij})$, and $|K|$ is the number of summed key terms. For example, based on the document set in Table 3-1, the derived membership functions are shown in Figure 3-3.

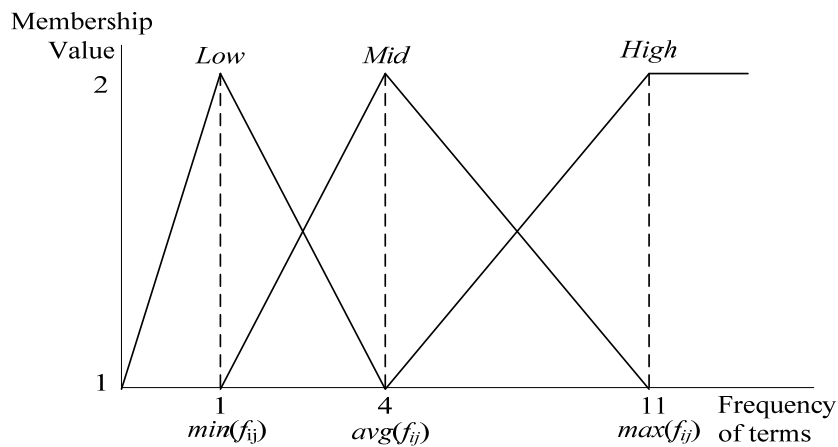


Figure 3-3: The predefined membership functions of this example.

3.2.2 The Fuzzy Association Rule Mining Algorithm for Text

To describe our fuzzy association rule mining algorithm shown, we need the Definitions 3.6 - 3.7. The candidate cluster set \tilde{C}_D for a document set D can be generated by Algorithm 3.2 shown in Figure 3-4 .

Definition 3.6: For a document set D , a *candidate cluster* $\tilde{c} = (\tilde{D}_c, \tau)$ is a two-tuple, where \tilde{D}_c is a subset of the document set D , such that it includes those documents which contain all the key terms in $\tau = \{t_1, t_2, \dots, t_q\} \subseteq K_D$, $q \geq 1$, where K_D is the key term set of D and q is the number of key terms included in τ . In fact, τ is a fuzzy frequent itemset for describing \tilde{c} . To illustrate, \tilde{c} can also be denoted as $\tilde{c}_{(t_1, t_2, \dots, t_q)}^q$ or $\tilde{c}_{(\tau)}^q$, and will be used interchangeably hereafter. For instance, in Table 3-1, the *candidate cluster* $\tilde{c}_{(\text{stock})}^1 = (\{d_1, d_2, d_3, d_8\}, \{\text{stock}\})$, as the term “stock” appeared in these documents.

Definition 3.7: The *candidate cluster set* of a document set D , denoted $\tilde{C}_D = \{\tilde{c}_1^1, \dots, \tilde{c}_{l-1}^2, \tilde{c}_l^q, \dots, \tilde{c}_k^q\}$, is a set of candidate clusters, where k is the total number of candidate clusters.

Algorithm 3.2. Basic algorithm to obtain the fuzzy frequent itemsets from the document set.

Input: A set of documents $D = \{d_1, d_2, \dots, d_m\}$, where $d_i = \{(t_1, f_{i1}), (t_2, f_{i2}), \dots, (t_j, f_{ij}), \dots, (t_m, f_{im})\}$; A set of membership functions (as defined in Section 3.2.1); The minimum support value θ ; The minimum confidence value λ .

Output: A set of candidate cluster.

1. For each $d_i \in D$ do
For each $t_j \in d_i$ do
(1) $f_{ij} \rightarrow F_{ij} = w_{ij}^{Low}/t_j.Low + w_{ij}^{Mid}/t_j.Mid + w_{ij}^{High}/t_j.High$ //using membership functions
2. For each $t_j \in K_D$ do
For each $d_i \in D$ do
(1) $count_j^{Low} = \sum_{i=1}^n w_{ij}^{Low}, count_j^{Mid} = \sum_{i=1}^n w_{ij}^{Mid}, count_j^{High} = \sum_{i=1}^n w_{ij}^{High}$
3. For each $t_j \in K_D$ do
(1) $max-count_j = \max(count_j^{Low}, count_j^{Mid}, count_j^{High})$
4. $L_1 = \{max-R_j | support(t_j) = \frac{max-count_j}{|D|} \geq \theta, 1 \leq j \leq m\}$ // $|D|$ is the number of documents.
5. For ($q = 2; L_{q-1} \neq \emptyset; q++$) do // Find fuzzy frequent q -itemsets L_q
 - (1) $C_q = \mathbf{apriori_gen}(L_{q-1}, \theta)$ // similar to the *a priori* algorithm
 - (2) For each candidate q -itemsets τ with key terms $(t_1, t_2, \dots, t_q) \in C_q$ do
 - (a) For each $d_i \in D$ do
 $w_{i\tau} = \min\{w_{ij}^{max-R_j} | j = 1, 2, \dots, q\}$ // $w_{ij}^{max-R_j}$ is the fuzzy membership value of the maximum region of t_j in d_i .
 - (b) $count_\tau = \sum_{i=1}^n w_{i\tau}$
 - (3) $L_q = \{\tau \in C_q | support(\tau) = \frac{count_\tau}{|D|} \geq \theta, 1 \leq j \leq q\}$
6. For all the fuzzy frequent q -itemsets τ containing key terms (t_1, t_2, \dots, t_q) , where $q \geq 2$ do // construct the strong fuzzy frequent itemsets
 - (1) form all possible association rules
 $\tau_1 \wedge \dots \wedge \tau_{k-1} \wedge \tau_{k+1} \wedge \dots \wedge \tau_q \rightarrow \tau_k, k = 1$ to q .
 - (2) Calculate the confidence values of all possible association rules
$$confidence(\tau) = \frac{\sum_{i=1}^n w_{i\tau}}{\sum_{i=1}^n (w_{i1} \wedge \dots \wedge w_{ik-1}, w_{ik+1} \wedge \dots \wedge w_{iq})}$$
 - (3) $\tilde{C}_D = \{\tau \in L_q | confidence(\tau) \geq \lambda\}$
7. $\tilde{C}_D \rightarrow \{L_1\} \cup \tilde{C}_D$

Procedure **apriori_gen**(L_{q-1}, θ)

1. for each itemset $l_1 \in L_{q-1}$ do
for each itemset $l_2 \in L_{q-1}$ do
(1) if ($l_1[1] = l_2[1] \ l_1[2] = l_2[2] \ \dots \ l_1[k-2] = l_2[k-2] \ l_1[k-1] = l_2[k-1]$) then
 $C_q = \{c | c = l_1 \ l_2\}$
 2. Return C_q
-

Figure 3-4: A detailed illustration of Algorithm 3-2.

3.2.3 An Illustrative Example of Stage 2

Consider using the document set D in Table 3-1, the membership functions defined in Figure 3-3, the minimum support value 40%, and the minimum confidence value 60% as inputs. The procedure of Algorithm 3.2 is illustrated in the following.

Step 1. The input membership functions are used to convert each term frequency into a fuzzy set. By taking the first key term t_1 “stock” in document d_1 as an example, its term frequency ‘2’ will be transformed into the fuzzy set $F_{11} = 1.67/stock.Low + 1.33/stock.Mid + 1.0/stock.High$ based on the given membership functions, where the notation term.region is called a fuzzy region. This step will be repeated for the other terms, and the results are shown in Table 3-2.

Table 3-2 : The fuzzy set in this example.

| Doc ID | Level-1 Fuzzy Set | | | | | | | | | | | | | | | | | |
|----------|-------------------|------|------|--------|------|------|--------|------|------|---------|------|------|-----------|------|------|--------|------|------|
| | stock | | | record | | | profit | | | medical | | | treatment | | | health | | |
| | L | M | H | L | M | H | L | M | H | L | M | H | L | M | H | L | M | H |
| d_1 | 1.67 | 1.33 | 1.00 | 2.00 | 1.00 | 1.00 | 2.00 | 1.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| d_2 | 2.00 | 1.00 | 1.00 | 2.00 | 1.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| d_3 | 2.00 | 1.00 | 1.00 | 0.00 | 0.00 | 0.00 | 1.67 | 1.33 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| d_4 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.33 | 1.67 | 1.00 | 0.00 | 0.00 | 0.00 | 1.67 | 1.33 | 1.00 |
| d_5 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 2.00 | 2.00 | 1.00 | 1.00 | 2.00 | 1.00 | 1.00 |
| d_6 | 0.00 | 0.00 | 0.00 | 2.00 | 1.00 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 | 2.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| d_7 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 1.43 | 1.57 | 2.00 | 1.00 | 1.00 | 1.67 | 1.33 | 1.00 |
| d_8 | 1.33 | 1.67 | 1.00 | 0.00 | 0.00 | 0.00 | 2.00 | 1.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| d_9 | 0.00 | 0.00 | 0.00 | 2.00 | 1.00 | 1.00 | 0.00 | 0.00 | 0.00 | 1.33 | 1.67 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| d_{10} | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 1.43 | 1.57 | 1.67 | 1.33 | 1.00 | 2.00 | 1.00 | 1.00 |

Step 2. For D , the scalar cardinality of each fuzzy region for each key term is calculated as count value. For example, the scalar cardinality of the fuzzy region $stock.Low = (1.67 + 2.00 + 2.00 + 1.33) = 7.0$. By repeating this step for the other regions, the results can be obtained as Table 3-3 illustrates.

Table 3-3: The count values of three fuzzy regions for each key term.

| Terms | Count | Terms | Count | Terms | Count |
|-------------|-------|--------------|-------|----------------|-------|
| stock.Low | 7.00 | profit.Low | 5.67 | treatment.Low | 5.67 |
| stock.Mid | 5.00 | profit.Mid | 3.33 | treatment.Mid | 3.33 |
| stock.High | 4.00 | profit.High | 3.00 | treatment.High | 3.00 |
| record.Low | 8.00 | medical.Low | 6.66 | health.Low | 7.34 |
| record.Mid | 4.00 | medical.Mid | 9.20 | health.Mid | 4.66 |
| record.High | 4.00 | medical.High | 8.14 | health.High | 4.00 |

Step 3. Then, the region of each key term with maximum count value will be found.

Take the key term “stock” as an example. Its count value is 7.0 for *Low*, 5.0 for *Mid*, and 4.0 for *High*. Due to the count value for *Low* is the highest among the three count values, the region *Low* is thus used to represent the key term “stock” in the following steps. This step is repeated for the other key terms. Thus, *Low* is chosen for “stock”, “record”, “profit”, “treatment”, and “health”, and *Mid* is chosen for “medical”.

Step 4. According to the maximum count value for each key term chosen in Step 3, these key terms must be checked against the predefined minimum support value 40%. Since the count values of stock.Low, record.Low, profit.Low, treatment.Low, medical.Mid, and health.Low, are all larger than 40%, these key terms are put in L_1 (fuzzy frequent 1-itemsets) as shown in Table 3-4.

Table 3-4 : The set of fuzzy frequent 1-itemsets in this example.

| Terms | Count | Support Values |
|---------------|-------|----------------|
| stock.Low | 7.00 | 7.00/10=70% |
| Record.Low | 8.00 | 8.00/10=80% |
| profit.Low | 5.67 | 5.67/10=57% |
| medical.Mid | 9.20 | 9.20/10=92% |
| treatment.Low | 5.67 | 5.67/10=57% |
| health.Low | 7.34 | 7.34/10=73% |

Step 5. (1)The candidate set C_2 is generated from L_1 as shown in Table 3-5.

Table 3-5 : The candidate set C_2 .

| Candidate 2-itemsets | Candidate 2-itemsets | Candidate 2-itemsets |
|----------------------------|-----------------------------|------------------------------|
| (stock.Low, record.Low) | (record.Low, profit.Low) | (profit.Low, treatment.Low) |
| (stock.Low, profit.Low) | (record.Low, medical.Mid) | (profit.Low, health.Low) |
| (stock.Low, medical.Mid) | (record.Low, treatment.Low) | (medical.Mid, treatment.Low) |
| (stock.Low, treatment.Low) | (record.Low, health.Low) | (medical.Mid, health.Low) |
| (stock.Low, health.Low) | (profit.Low, medical.Mid) | (treatment.Low, health.Low) |

(2) For each candidate 2-itemset in C_2 , there are three sub-steps to be performed:

(a) The fuzzy value of each document for each candidate 2-itemset is calculated. For instance, the derived fuzzy value of (stock.Low, record.Low) in document d_1 can be calculated as: $\min(1.67, 2.00) = 1.67$. The results for the other documents are shown in Table 3-6.

Table 3-6 : The fuzzy values of (stock.Low, record.Low) in D .

| DocID | stock.Low | record.Low | $\min(\text{stock.Low}, \text{record.Low})$ |
|----------|-----------|------------|---|
| d_1 | 1.67 | 2.00 | 1.67 |
| d_2 | 2.00 | 2.00 | 2.00 |
| d_3 | 2.00 | 0.00 | 0.00 |
| d_4 | 0.00 | 0.00 | 0.00 |
| d_5 | 0.00 | 0.00 | 0.00 |
| d_6 | 0.00 | 2.00 | 0.00 |
| d_7 | 0.00 | 0.00 | 0.00 |
| d_8 | 1.33 | 0.00 | 0.00 |
| d_9 | 0.00 | 2.00 | 0.00 |
| d_{10} | 0.00 | 0.00 | 0.00 |

(b) Calculate the scalar cardinality for each candidate 2-itemset. Table 3-8 lists the results for all candidate 2-itemsets.

Table 3-7 : The count values of candidate 2-itemsets.

| Candidate 2-itemsets | Count | Support Values |
|------------------------------|-------|----------------|
| (stock.Low, record.Low) | 3.67 | 3.67/10=37% |
| (stock.Low, profit.Low) | 4.67 | 4.67/10=47% |
| (stock.Low, medical.Mid) | 0.00 | 0% |
| (stock.Low, treatment.Low) | 0.00 | 0% |
| (stock.Low, health.Low) | 0.00 | 0% |
| (record.Low, profit.Low) | 2.00 | 2.00/10=20% |
| (record.Low, medical.Mid) | 3.67 | 3.67/10=37% |
| (record.Low, treatment.Low) | 0.00 | 0% |
| (record.Low, health.Low) | 0.00 | 0% |
| (profit.Low, medical.Mid) | 0.00 | 0% |
| (profit.Low, treatment.Low) | 0.00 | 0% |
| (profit.Low, health.Low) | 0.00 | 0% |
| (medical.Mid, treatment.Low) | 3.86 | 3.86/10=39% |
| (medical.Mid, health.Low) | 5.53 | 5.53/10=55% |
| (treatment.Low, health.Low) | 5.34 | 5.34/10=53% |

- (3) Because only the count values of (stock.Low, profit.Low), (medical.Mid, health.Low), and (treatment.Low, health.Low) are larger than the predefined minimum support value 40%. Thus, they are stored in L_2 (fuzzy frequent 2-itemsets).

Step 5. Since L_2 is not null, repeat the step 5 as follows.

- (1) q , a variable used to store the number of key terms kept in the current itemsets, is set as 2.
- (2) The candidate 3-itemset (medical.Mid, health.Low, treatment.Low) is generated from L_2 . The count value of the candidate 3-itemset (medical.Mid, health.Low, treatment.Low) is 3.00.
- (3) Then, its support value is $3.00/10 = 0.30$. Since its support value is not larger than 40%, it is not put in L_3 .

Step 6. Since L_3 is null, we proceed to step 6. For each fuzzy frequent itemset, the

association rules are constructed by accomplishing the following sub-steps.

- (a) Based on the fuzzy frequent itemsets, all possible association rules are formed:

If stock = *Low*, then profit = *Low*

If profit = *Low*, then stock = *Low*

If medical = *Mid*, then health = *Low*

If health = *Low*, then medical = *Mid*

If treatment = *Low*, then health = *Low*

If health = *Low*, then treatment = *Low*

- (b) Then, we calculate the confidence values of the above possible association rules. Take the first rule pair as an example. Their confidence values are calculated as follows:

- If stock = *Low*, then profit = *Low*

$$\frac{\sum_{i=1}^{10} (\text{stock.Low} \cap \text{profit.Low})}{\sum_{i=1}^{10} (\text{stock.Low})} = \frac{4.67}{7.0} = 67\%$$

- If profit = *Low*, then stock = *Low*

$$\frac{\sum_{i=1}^{10} (\text{stock.Low} \cap \text{profit.Low})}{\sum_{i=1}^{10} (\text{profit.Low})} = \frac{4.67}{5.67} = 82\%$$

For the other rule pairs, the results are shown below:

If medical = *Mid*, then health = *Low*, with a confidence value of 0.60.

If health = *Low*, then Medical = *Mid*, with a confidence value of 0.75.

If treatment = *Low*, then health = *Low*, with a confidence value of 0.94.

If health = *Low*, then treatment = *Low*, with a confidence value of 0.73.

In the proposed algorithm, we estimate the strength of association among key terms in the document set by using confidence values. There is useful information when the co-occurring keywords have been shown. This is because highly co-occurring terms are used together. Thus, our algorithm compute the confidence values of a rule pair to check the strong association of key terms (t_1, t_2, \dots, t_q) of the fuzzy frequent q -itemsets. Take the candidate cluster $\tilde{c}_{(\text{stock}, \text{profit})}^2$ as an example. Since its confidence value of the rule pair “If stock = *Low*, then profit = *Low*” and “If profit = *Low*, then stock = *Low*” are both larger than the minimum confidence value 60%, $\tilde{c}_{(\text{stock}, \text{profit})}^2$ is put in the candidate cluster set \tilde{C}_D . Finally, the candidate cluster set $\tilde{C}_D = \{ \tilde{c}_{(\text{stock})}^1, \tilde{c}_{(\text{record})}^1, \tilde{c}_{(\text{profit})}^1, \tilde{c}_{(\text{medical})}^1, \tilde{c}_{(\text{treatment})}^1, \tilde{c}_{(\text{health})}^1, \tilde{c}_{(\text{stock}, \text{profit})}^2, \tilde{c}_{(\text{medical}, \text{health})}^2, \tilde{c}_{(\text{treatment}, \text{health})}^2 \}$ will be output.

3.3 Stage 3: The Cluster Tree Construction

The candidate cluster set generated by the previous steps can be viewed as a set of topics with their corresponding sub-topics in the document set. In this stage, we first construct the Document-Term Matrix (DTM) and the Term-Cluster Matrix (TCM) to derive the Document-Cluster matrix (DCM) for assigning each document to a fitting cluster, such that each c_i^q contains a subset of documents. For the documents in each c_i^q , the intra-cluster similarity is minimized and the inter-clusters similarity is maximized. We call each c_i^q a *target cluster* in the following. Based on the assignment result, we will find the set of target clusters $C_D = \{c_1^1, c_2^1, \dots, c_i^q, \dots, c_f^q\}$, and then use these target clusters to form a hierarchical tree for the document set D .

To avoid the constructed cluster tree including too many clusters, we use the tree pruning method to prune unnecessary clusters.

3.3.1 Building the Document-Cluster Matrix (DCM)

First, consider each candidate cluster $\tilde{c}_{(\tau)}^q = \tilde{c}_{(t_1, t_2, \dots, t_q)}^q$ with fuzzy frequent itemset τ . τ will be regarded as a reference point for generating a target cluster. Then, to represent the degree of importance of document d_i in a candidate cluster \tilde{c}_i^q , an $n \times k$ Document-Cluster Matrix will be constructed to calculate the similarity of terms in d_i and \tilde{c}_i^q . To achieve this goal, we have to define two matrixes in Definition 3.8 and Definition 3.9, namely Document-Term Matrix and Term-Cluster Matrix, respectively. Finally, based on Definitions 3.8 - 3.9, we can define the Document-Cluster Matrix (DCM) of a document set D in Definition 3.10.

Definition 3.8: A *Document-Term Matrix (DTM)*, denoted $W = [w_{ij}^{\max-R_j}]$, for a document set D , is an $n \times p$ matrix, such that $w_{ij}^{\max-R_j}$ is the weight (fuzzy membership value of the maximum region) of term t_j in document d_i and $t_j \in L_1$ and can be calculated from the Steps 4 and 5 of Algorithm 3-2. A formal illustration of DTM can be found in Figure 3-5.

$$W = \begin{matrix} & \begin{matrix} t_1 & t_2 & \dots & t_p \end{matrix} \\ \begin{matrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{matrix} & \begin{bmatrix} w_{11}^{\max-R_j} & w_{12}^{\max-R_j} & \dots & w_{1p}^{\max-R_j} \\ w_{21}^{\max-R_j} & w_{22}^{\max-R_j} & \dots & w_{2p}^{\max-R_j} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1}^{\max-R_j} & w_{n2}^{\max-R_j} & \dots & w_{np}^{\max-R_j} \end{bmatrix} \end{matrix} \Big]_{n \times p}$$

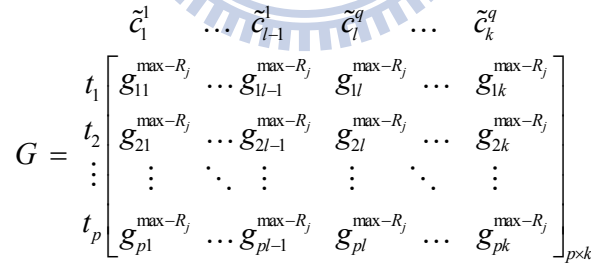
Figure 3-5: A formal illustration of Document-Term Matrix.

Definition 3.9: A *Term-Cluster Matrix (TCM)*, denoted $G = [g_{jl}^{\max-R_j}]$, for a document set D of n documents, is an $p \times k$ matrix, such that for $1 \leq j \leq p$, $1 \leq l \leq k$, and

$$g_{jl}^{\max-R_j} = \frac{\text{score}(\tilde{c}_l^q)}{\sum_{i=1}^n w_{ij}^{\max-R_j}}, \text{ where } \text{score}(\tilde{c}_l^q) = \left. \begin{array}{l} \sum_{d_i \in \tilde{c}_l^1, t_j \in L_1} w_{ij}^{\max-R_j} \quad \text{if } q = 1, \\ \frac{\sum_{d_i \in \tilde{c}_l^q, t_j \in L_1} w_{ij}^{\max-R_j}}{\lambda} \quad \text{else,} \end{array} \right\}. \quad (3.7)$$

In Formula (3.7), $w_{ij}^{\max-R_j}$ is the weight (fuzzy membership value of the maximum region) of term t_j in document $d_i \in \tilde{c}_l^q$ and λ is the minimum confidence value.

Each $g_{jl}^{\max-R_j}$ of TCM represents the degree of importance of key term t_j in a candidate cluster $\tilde{c}_{(\tau)}^q$ by referring to those documents including τ . To reduce the dimension, only key terms appeared in L_1 were applied in TCM. A formal illustration of TCM can be found in Figure 3-6.



$$G = \begin{matrix} & \tilde{c}_1^1 & \dots & \tilde{c}_{l-1}^1 & \tilde{c}_l^q & \dots & \tilde{c}_k^q \\ \begin{matrix} t_1 \\ t_2 \\ \vdots \\ t_p \end{matrix} & \begin{bmatrix} g_{11}^{\max-R_j} & \dots & g_{1l-1}^{\max-R_j} & g_{1l}^{\max-R_j} & \dots & g_{1k}^{\max-R_j} \\ g_{21}^{\max-R_j} & \dots & g_{2l-1}^{\max-R_j} & g_{2l}^{\max-R_j} & \dots & g_{2k}^{\max-R_j} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ g_{p1}^{\max-R_j} & \dots & g_{pl-1}^{\max-R_j} & g_{pl}^{\max-R_j} & \dots & g_{pk}^{\max-R_j} \end{bmatrix} & \end{matrix} \Big]_{p \times k}$$

Figure 3-6: A formal illustration of Term-Cluster Matrix.

Definition 3.10: A *Document-Cluster Matrix (DCM)* for a document set D of n documents is the inner product of its DTM and TCM. It is an $n \times k$ matrix, and can be defined as $V = [v_{il}]$, where

$$v_{il} = \text{row}_i(W) \cdot \text{col}_l(G) = \begin{bmatrix} w_{i1}^{\max-R_j} & w_{i2}^{\max-R_j} & \dots & w_{ip}^{\max-R_j} \end{bmatrix} \begin{bmatrix} g_{1l}^{\max-R_j} \\ g_{2l}^{\max-R_j} \\ \vdots \\ g_{pl}^{\max-R_j} \end{bmatrix} = \sum_{p=1}^p w_{ip} g_{pl}^{\max-R_j}, \quad 1 \leq i \leq n \text{ and } 1 \leq l \leq k.$$

A formal illustration of DCM can be found in Figure 3-7.

$$V = \begin{matrix} & \tilde{c}_{11}^1 \dots & \tilde{c}_{1l-1}^2 & \tilde{c}_{1l}^q & \dots & \tilde{c}_{1k}^q \\ d_1 & \begin{bmatrix} v_{11} & \dots & v_{1l-1} & v_{1l} & \dots & v_{1k} \end{bmatrix} \\ d_2 & \begin{bmatrix} v_{21} & \dots & v_{2l-1} & v_{2l} & \dots & v_{2k} \end{bmatrix} \\ \vdots & \begin{bmatrix} \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \end{bmatrix} \\ d_n & \begin{bmatrix} v_{n1} & \dots & v_{nl-1} & v_{nl} & \dots & v_{nk} \end{bmatrix} \\ & n \times k \end{matrix} = \begin{matrix} & t_1 & t_2 & \dots & t_p \\ d_1 & \begin{bmatrix} w_{21}^{\max-R_j} & w_{22}^{\max-R_j} & \dots & w_{2p}^{\max-R_j} \end{bmatrix} \\ d_2 & \begin{bmatrix} w_{21}^{\max-R_j} & w_{22}^{\max-R_j} & \dots & w_{2p}^{\max-R_j} \end{bmatrix} \\ \vdots & \begin{bmatrix} \vdots & \vdots & \dots & \vdots \end{bmatrix} \\ d_n & \begin{bmatrix} w_{21}^{\max-R_j} & w_{22}^{\max-R_j} & \dots & w_{2p}^{\max-R_j} \end{bmatrix} \\ & n \times p \end{matrix} \cdot \begin{matrix} & \tilde{c}_1^1 & \tilde{c}_2^1 & \dots & \tilde{c}_k^q \\ t_1 & \begin{bmatrix} g_{12}^{\max-R_j} \\ g_{22}^{\max-R_j} \\ \vdots \\ g_{p2}^{\max-R_j} \end{bmatrix} \\ t_2 & \begin{bmatrix} g_{12}^{\max-R_j} \\ g_{22}^{\max-R_j} \\ \vdots \\ g_{p2}^{\max-R_j} \end{bmatrix} \\ \vdots & \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} \\ t_p & \begin{bmatrix} g_{12}^{\max-R_j} \\ g_{22}^{\max-R_j} \\ \vdots \\ g_{p2}^{\max-R_j} \end{bmatrix} \\ & p \times k \end{matrix}$$

Figure 3-7: A formal illustration of Document-Cluster Matrix.

3.3.2 Building the Hierarchical Cluster Tree

Based on the obtained DCM, each document can be assigned to only one target cluster by using the following rules.

Rule 1. Each element v_{il} of the DCM matrix represents the degree of importance of document d_i in a candidate cluster \tilde{c}_l^1 . For each document d_i (the row i of DCM), if there exists only one maximum v_{il} in $\{v_{i1}, v_{i2}, \dots, v_{iy}\} \in \tilde{c}_{(\tau)}^1$ (the column 1 to y of DCM), where $1 \leq y \leq k$, then d_i will be assigned to a target cluster c_l^1 ; otherwise, apply Rule 2.

$$c_l^1 = \{d_i \mid v_{il} = \max\{v_{i1}, v_{i2}, \dots, v_{iy}\} \in \tilde{c}_{(\tau)}^1, \text{ where } 1 \leq y \leq k\} \quad (3.8)$$

Rule 2. If a document d_i has the same maximum values $\{v_{i1}, v_{i2}, \dots, v_{iy}\} \in \tilde{c}_{(\tau)}^1$ from more than one of the candidate clusters $\{\tilde{c}_1^1, \tilde{c}_2^1, \dots, \tilde{c}_y^1\}$, then d_i will be

assigned to a target cluster c_i^1 , such that its fuzzy frequent itemset τ has the highest count value. Notice that when $q = 1$, the count value is *max-count_i* (refer to the Step 3 in Algorithm 3-2).

After assigning each document to the best fitting cluster, the resulting tree can be formed as a foundation for pruning and a natural structure for browsing. The cluster tree built by F²IHC algorithm has the following eight features:

1. The cluster tree is built in a top-down fashion, which is different from the cluster tree obtained in a bottom-up fashion by FIHC.
2. Each child target cluster has exactly one parent target cluster.
3. The topic of a parent target cluster is more general than the topic of its children target clusters. The nodes become more and more specialized as they get closer to the leaf nodes.
4. A parent target cluster and its children target clusters are “similar” to a certain degree.
5. Each target cluster employs one fuzzy frequent q -itemset τ as its cluster label.
6. The root node of the cluster tree appears at level 0, and is tagged with the cluster label “all”.
7. Each target cluster with its fuzzy frequent q -itemset appears in the level q of the tree.
8. The depth of the cluster tree is the same as the maximum size of fuzzy frequent itemsets.

3.3.3 Tree Pruning

When a low minimum support value and a low minimum confidence value are

used, the target cluster tree would become broad and deep. The documents with the same topic may be spread to several small target clusters, which would cause low document clustering accuracy. In order to generate a natural hierarchical cluster tree for higher document clustering accuracy and for easy browsing, one tree pruning method is used for merging similar target clusters at level 1. This method employs Definition 3.11 to compute the inter-cluster similarity between two target clusters. In the following, the minimum *Inter-Sim* will be used as a threshold δ to decide whether two target clusters should be merged.

Definition 3.11: The *inter-cluster similarity* between two target clusters c_x^1 and c_y^1 , $c_x^1 \neq c_y^1$, is defined by Formula (3.9):

$$Inter_Sim(c_x^1, c_y^1) = \frac{\sum_{d_i \in c_x^1, c_y^1} v_{ix} \times v_{iy}}{\sqrt{\sum_{d_i \in c_x^1} (v_{ix})^2 \times \sum_{d_i \in c_y^1} (v_{iy})^2}} \quad (3.9)$$

where v_{ix} and v_{iy} stand for two entry, such that $d_i \in c_x^1$, $d_i \in c_y^1$, in DCM, respectively; The range of *Sim* is $[0, 1]$. If the *Inter-Sim* value is close to 1, then both clusters are regarded nearly the same.

The objective of sibling merging is to shrink a tree by merging similar target clusters at level 1 for attaining high document clustering accuracy. Each pair of target clusters at level 1 of a tree is calculated by using the inter-cluster similarity measure. The target cluster pair with the highest *Inter-Sim* value is to keep merging until the *Inter-Sim* value of all target clusters at level 1 is less than the minimum *Inter-Sim* threshold δ .

Algorithm 3.3 shown in Figure 3-8 is used to assign each document to the best fitting cluster, and finally builds a cluster tree for output.

Algorithm 3.3. The cluster tree construction algorithm for building a hierarchical cluster tree for assigning each document to a fitting cluster.

Input: A document set $D = \{d_1, d_2, \dots, d_i, \dots, d_n\}$; The key term set $K_D = \{t_1, t_2, \dots, t_j, \dots, t_m\}$; The candidate cluster set $\tilde{C}_D = \{\tilde{c}_1^1, \dots, \tilde{c}_{l-1}^1, \tilde{c}_l^q, \dots, \tilde{c}_k^q\}$; A minimum *Inter-Sim* threshold δ .

Output: A cluster tree $CT = \{c_1^1, c_2^1, \dots, c_l^q, \dots, c_f^q\}$

1. Build $n \times p$ document-term matrix $W = [w_{ij}^{\max-R_j}]$
 2. Build $p \times k$ term-cluster matrix $G = [g_{jl}^{\max-R_j}]$.
 3. Build $n \times k$ document-cluster matrix $V = W \cdot G = [v_{il}] = \sum_{p=1}^p w_{ip} g_{pl}$.
 4. Assign a document to a best target cluster.
 - (1) $c_l^1 = \{d_i \mid v_{il} = \max\{v_{i1}, v_{i2}, \dots, v_{il}\} \in \tilde{c}_l^1, \text{ where the number of } v_{il} \text{ is } 1\}$; Otherwise, apply Rule 2
 - (2) $c_l^1 = \{d_i \mid v_{il} = \max\{v_{i1}, v_{i2}, \dots, v_{il}\} \in \tilde{c}_l^1, \text{ where the number of } v_{il} > 1 \text{ and with } \tilde{c}_l^1 \text{ the highest fuzzy count value } \max\text{-count}_l \text{ corresponding to its fuzzy frequent itemset}\}$.
 5. Sibling merging
 - (1) Remove the empty node at level 1.
 - (a) If there is no documents in target cluster c_l^1 then $\{\text{this empty target cluster } c_l^1 \text{ skip, and cluster } c_{l+1}^1 \text{ try}\}$.
 - (2) For each pair of target clusters at level 1.
 - (a) Calculate the *Inter_Sim*.
 - (b) Store the results in the inter-cluster similarity matrix I .
 - (3) Keeping merging until the *Inter_Sim* of each pair of target clusters at level 1 is less than δ .
 - (a) Select the target cluster pair with the highest *Inter_Sim*.
 - (b) Merge the smaller target cluster into the larger target cluster.
 - (c) Repeat Step 5(2) to update I .
 6. Tree construction.
 - (1) Sort all target clusters by the number of key terms with its fuzzy frequent q -itemset in ascending order.
 - (2) Remove the candidate clusters that have no parent target clusters.
 - (3) Identify all potential children:
 - (a) Let q be the number of terms in c_l^q 's fuzzy frequent q -itemset.
 - (b) *PotentialChildren* = Find all target clusters with $q + 1$ terms, such that each of the key term is a subset of c_l^q 's key terms in fuzzy frequent q -itemset.
 - (4) Choose the most similar children:
 - (a) Find a parent target cluster and its children target clusters against each *PotentialChildren*.
 - (b) Set the potential children cluster.
 - (5) Children splitting.
 - (a) Scan the tree in a top-down fashion.
 - (b) For each non-leaf node c_l^q in level q of the tree do
 - For each document in c_l^q do
 - Based on DCM, if the v_{il} value of a document d_i in parent cluster c_l^{q-1} is equal to or lower than that of some of its children cluster c_l^q ; then this document will be moved to the child cluster with the maximum v_{il} value.
 7. Output the cluster tree CT .
-

Figure 3-8: A detailed illustration of Algorithm 3.3.

3.3.4 An Illustrative Example of Stage 3

For example, consider the document set in Table 3-1. The key term set $K_D = \{\text{stock, record, profit, medical, treatment, health}\}$, which was generated in Section 3.2.1. The candidate cluster set $\tilde{C}_D = \{\tilde{c}_{(\text{stock})}^1, \tilde{c}_{(\text{record})}^1, \tilde{c}_{(\text{profit})}^1, \tilde{c}_{(\text{medical})}^1, \tilde{c}_{(\text{treatment})}^1, \tilde{c}_{(\text{health})}^1, \tilde{c}_{(\text{stock, profit})}^2, \tilde{c}_{(\text{medical, health})}^2, \tilde{c}_{(\text{treatment, health})}^2\}$ was already generated in Section 3.2.3.

Now, suppose the minimum *Inter-Sim* value is 0.6. The proposed cluster tree construction algorithm proceeds as follows:

Step 1. Build 10×6 Document-Term matrix W in Table 3-8.

Table 3-8: The DTM of this example.

| Documents/ Key Terms | stock.Low | record.Low | profit.Low | medical.Mid | treatment.Low | health.Low |
|---------------------------------|-----------|------------|------------|-------------|---------------|------------|
| d_1 | 1.67 | 2.00 | 2.00 | 0.00 | 0.00 | 0.00 |
| d_2 | 2.00 | 2.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| d_3 | 2.00 | 0.00 | 1.67 | 0.00 | 0.00 | 0.00 |
| d_4 | 0.00 | 0.00 | 0.00 | 1.67 | 0.00 | 1.67 |
| d_5 | 0.00 | 0.00 | 0.00 | 1.00 | 2.00 | 2.00 |
| d_6 | 0.00 | 2.00 | 0.00 | 2.00 | 0.00 | 0.00 |
| d_7 | 0.00 | 0.00 | 0.00 | 1.43 | 2.00 | 1.67 |
| d_8 | 1.33 | 0.00 | 2.00 | 0.00 | 0.00 | 0.00 |
| d_9 | 0.00 | 2.00 | 0.00 | 1.67 | 0.00 | 0.00 |
| d_{10} | 0.00 | 0.00 | 0.00 | 1.43 | 1.67 | 2.00 |

Step 2. Build 6×9 Term-Cluster matrix G in Table 3-9.

Table 3-9: The TCM of this example.

| Key Terms / Clusters | $\tilde{c}_{(stock)}^1$ | $\tilde{c}_{(record)}^1$ | $\tilde{c}_{(profit)}^1$ | $\tilde{c}_{(medical)}^1$ | $\tilde{c}_{(treatment)}^1$ | $\tilde{c}_{(health)}^1$ | $\tilde{c}_{(stock, profit)}^2$ | $\tilde{c}_{(medical, health)}^2$ | $\tilde{c}_{(treatment, health)}^2$ |
|---------------------------------|-------------------------|--------------------------|--------------------------|---------------------------|-----------------------------|--------------------------|---------------------------------|-----------------------------------|-------------------------------------|
| stock.Low | 1.00 | 0.52 | 0.71 | 0.00 | 0.00 | 0.00 | 1.19 | 0.00 | 0.00 |
| record.Low | 0.50 | 1.00 | 0.25 | 0.50 | 0.00 | 0.00 | 0.42 | 0.00 | 0.00 |
| profit.Low | 1.00 | 0.35 | 1.00 | 0.00 | 0.00 | 0.00 | 1.67 | 0.00 | 0.00 |
| medical.Mid | 0.00 | 0.40 | 0.00 | 1.00 | 0.42 | 0.60 | 0.00 | 1.00 | 0.70 |
| treatment.Low | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 | 0.00 | 1.67 | 1.67 |
| health.Low | 0.00 | 0.00 | 0.00 | 1.00 | 0.77 | 1.00 | 0.00 | 1.67 | 1.29 |

Step 3. Build 10×9 Document-Cluster matrix V in Table 3-10.

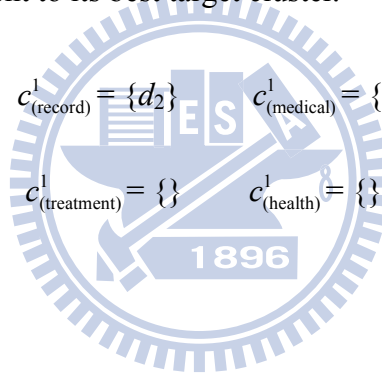
Table 3-10: The DCM of this example.

| Documents / Clusters | $\tilde{c}_{(\text{stock})}^1$ | $\tilde{c}_{(\text{record})}^1$ | $\tilde{c}_{(\text{profit})}^1$ | $\tilde{c}_{(\text{medical})}^1$ | $\tilde{c}_{(\text{treatment})}^1$ | $\tilde{c}_{(\text{health})}^1$ | $\tilde{c}_{(\text{stock, profit})}^2$ | $\tilde{c}_{(\text{medical, health})}^2$ | $\tilde{c}_{(\text{treatment, health})}^2$ |
|----------------------|--------------------------------|---------------------------------|---------------------------------|----------------------------------|------------------------------------|---------------------------------|--|--|--|
| d_1 | 4.67 | 3.58 | 3.69 | 1.00 | 0.00 | 0.00 | 6.15 | 0.00 | 0.00 |
| d_2 | 3.00 | 3.05 | 1.93 | 1.00 | 0.00 | 0.00 | 3.21 | 0.00 | 0.00 |
| d_3 | 3.67 | 1.64 | 3.10 | 0.00 | 0.00 | 0.00 | 5.16 | 0.00 | 0.00 |
| d_4 | 0.00 | 0.67 | 0.00 | 3.34 | 1.99 | 2.67 | 0.00 | 4.46 | 3.32 |
| d_5 | 0.00 | 0.40 | 0.00 | 5.00 | 3.96 | 4.60 | 0.00 | 7.67 | 6.61 |
| d_6 | 1.00 | 2.80 | 0.50 | 3.00 | 0.84 | 1.20 | 0.83 | 2.00 | 1.40 |
| d_7 | 0.00 | 0.57 | 0.00 | 5.10 | 3.89 | 4.53 | 0.00 | 7.55 | 6.48 |
| d_8 | 3.33 | 1.40 | 2.95 | 0.00 | 0.00 | 0.00 | 4.92 | 0.00 | 0.00 |
| d_9 | 1.00 | 2.67 | 0.50 | 2.67 | 0.70 | 1.00 | 0.83 | 1.67 | 1.17 |
| d_{10} | 0.00 | 0.57 | 0.00 | 5.10 | 3.81 | 4.53 | 0.00 | 7.55 | 6.36 |

* Numbers appeared in boldface mean the largest values of each row of $\tilde{c}_{(\tau)}^1$.

Step 4. Assign each document to its best target cluster.

$$\begin{aligned}
 c_{(\text{stock})}^1 &= \{d_1, d_3, d_8\} & c_{(\text{record})}^1 &= \{d_2\} & c_{(\text{medical})}^1 &= \{d_4, d_5, d_6, d_7, d_9, d_{10}\} \\
 c_{(\text{profit})}^1 &= \{\} & c_{(\text{treatment})}^1 &= \{\} & c_{(\text{health})}^1 &= \{\}
 \end{aligned}$$



Step 5. Sibling merging.

(1) Remove the empty node $\{c_{(\text{profit})}^1, c_{(\text{treatment})}^1, c_{(\text{health})}^1\}$.

(2) The *Inter_Sim* values of all pairs of target clusters are calculated in Table 3-11.

Table 3-11: The *Inter_Sim* values of all target clusters.

| Cluster pairs (c_x, c_y) | Inter_Sim |
|---|-----------|
| $(c_{(\text{stock})}, c_{(\text{record})})$ | 0.94 |
| $(c_{(\text{stock})}, c_{(\text{medical})})$ | 0.14 |
| $(c_{(\text{record})}, c_{(\text{medical})})$ | 0.34 |

(3) Keep merging until the *Inter_Sim* of all pairs of target clusters are lower than the minimum *Inter-Sim* value 0.6.

- (a) Based on the above result, the cluster pair $(c_{(\text{stock})}, c_{(\text{record})})$ has the highest *Inter_Sim* value.
- (b) Since the number of documents of $c_{(\text{record})}^1$ is less than $c_{(\text{stock})}^1$, the document in $c_{(\text{record})}^1$ is merged into $c_{(\text{stock})}^1$. Thus, $c_{(\text{stock})}^1 = \{d_1, d_2, d_3, d_8\}$.
- (c) Update the inter-cluster similarity matrix. We omit the details here due to space limitation.

Step 6. Tree construction.

- (1) Sort all target clusters based on the number of key terms, we obtain

$$\{ c_{(\text{stock})}^1, c_{(\text{medical})}^1, c_{(\text{stock, profit})}^2, c_{(\text{medical, health})}^2, c_{(\text{treatment, health})}^2 \}.$$

- (2) Remove the target clusters and it has no parent clusters to produce the result $\{ c_{(\text{treatment, health})}^2 \}$.

- (3) Identify all potential children.

- (a) The number of terms in $c_{(\text{stock, profit})}^2$ and $c_{(\text{medical, health})}^2$ are both 2.

- (b) The *PotentialChildren* of $c_{(\text{stock})}^1$ is $c_{(\text{stock, profit})}^2$ and the

$$\textit{PotentialChildren} \text{ of } c_{(\text{medical})}^1 \text{ is } c_{(\text{medical, health})}^2.$$

- (4) The target clusters $c_{(\text{stock, profit})}^2$ and $c_{(\text{medical, health})}^2$ are set as the child cluster of $c_{(\text{stock})}^1$ and $c_{(\text{medical})}^1$, respectively.

- (5) Children splitting.

- (a) Here, we take the documents in the parent cluster $c_{(\text{medical})}^1$ for example.

(b) Based on DCM, we compare the value v_{il} of each document in the parent cluster $c_{(\text{medical})}^1$ and its child cluster $c_{(\text{medical, health})}^2$ to decide whether the document is divided into the child cluster. The result is shown in Table 3-12.

Table 3-12: The compare results between the parent cluster $c_{(\text{medical})}^1$ and its child cluster $c_{(\text{medical, health})}^2$.

| Documents / Clusters | $\tilde{c}_{(\text{medical})}^1$ | $\tilde{c}_{(\text{medical, health})}^2$ | Whether the document is divided into the child cluster |
|----------------------|----------------------------------|--|--|
| d_4 | 3.34 | 4.46 | Yes |
| d_5 | 5.00 | 7.67 | Yes |
| d_6 | 3.00 | 2.00 | No |
| d_7 | 5.10 | 7.55 | Yes |
| d_9 | 2.67 | 1.67 | No |
| d_{10} | 5.10 | 7.55 | Yes |

* Numbers appeared in boldface mean the largest values of each row.

Step 7. Finally, the derived cluster tree CT can be shown in Figure 3-9.

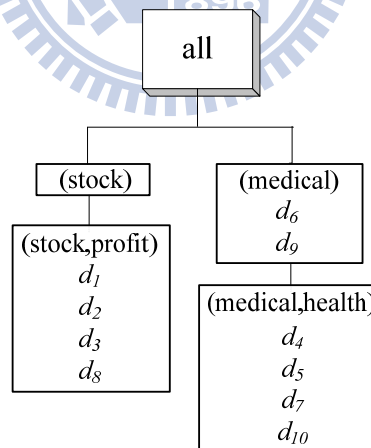


Figure 3-9: The derived hierarchical cluster tree.

3.4 Experiments

In this section, we experimentally evaluate the performance of the proposed algorithm by comparing with that of the FIHC approach. We make use of the FIHC 1.0 tool³ to generate the results of FIHC. The produced results are then fetched into the same evaluation program to ensure a fair comparison. All the experiments have been performed on a P4 3.2GHz Windows XP machine with 1GB memory.

3.4.1 Datasets

We used the five standard datasets employed by the FIHC experiments. These datasets are widely adopted as standard benchmarks for the text categorization task. To find key terms, stop words were removed and stemming was performed. Documents then were represented as TF (Term Frequency) vectors, and unimportant terms were discarded. This process implies a significant dimensionality reduction without loss of clustering performance.

The statistics of these datasets, after the document preprocessing described in Section 3.2.1, are summarized in Table 3-13. They are heterogeneous in terms of document size, cluster size, number of classes, and document distribution. The smallest document set contains 1,504 documents, and the largest one contains 8,649 documents. Each document is pre-classified into a single topic, i.e., a natural class. The class information is utilized in the evaluation method for measuring the accuracy

³ <http://ddm.cs.sfu.ca/dmssoft/Clustering/products/>

of the clustering result. The detailed information [42] of these datasets can be described as follow:

- *Classic*⁴: This dataset is a combination of the four classes CACM, CISI, CRAN, and MED abstracts. Classic includes 3,204 CACM documents, 1,460 CISI documents from information retrieval papers, 1,398 CRANFIELD documents from aeronautical system papers, and 1,033 MEDLINE documents from medical journals.
- *Hitech*: The *Hitech* dataset was derived from the San Jose Mercury newspaper articles, which are delivered as part of the Text REtrieval Conference⁵ (TREC) collection. The classes of this dataset are computers, electronics, health, medical, research, and technology.
- *Re0*: *Re0* is a dataset, derived from *Reuters-21578*⁶ text categorization test collection Distribution 1.0. *Re0* includes 1,504 documents belonging to 13 different classes.
- *Reuters*⁵: This dataset is extracted from newspaper articles. These documents are divided into 135 topics mostly concerning business and economy. In our test, we discarded documents with multiple category labels, and the result is consisting of documents associated with a single topic of approximately 9,000 documents and 50 classes. This dataset is also highly skewed.
- *Wap*: This dataset consists of 1,560 Web pages from Yahoo! Subject hierarchy collected and classified into 20 different classes for the WebACE project [19]. Many classes of *Wap* are close to each other.

⁴ <ftp://ftp.cs.cornell.edu/pub/smart/>

⁵ <http://trec.nist.gov/>

⁶ <http://www.daviddlewis.com/resources/testcollections/>

Table 3-13: Statistics for our test datasets.

| Datasets | Number of Documents | Number of Natural Clusters | Class Size | | | The Length of Documents |
|----------|---------------------|----------------------------|------------|---------|------|-------------------------|
| | Total | Total | Max | Average | Min | Average |
| Classic | 7094 | 4 | 3203 | 1774 | 1033 | 43 |
| Hitech | 2301 | 6 | 603 | 384 | 116 | 221 |
| Re0 | 1504 | 13 | 608 | 116 | 11 | 76 |
| Reuters | 8649 | 65 | 3725 | 131 | 1 | 42 |
| Wap | 1560 | 20 | 341 | 78 | 5 | 216 |

3.4.2 Evaluation of Cluster Quality: Overall F-measure

The *F-measure* is often employed to evaluate the accuracy of the generated clustering results. It is a standard evaluation method for both flat and hierarchical clustering structure. More importantly, this measure balances the cluster precision and cluster recall. Hence, we define a set of document clusters generated from the clustering result, denoted C , and another set, denoted L , consisting of natural classes, such as each document is pre-classified into a single class. Both sets are derived from the same document set D . Let $|D|$ be the number of all documents in the document set D ; $|c_i|$ be the number of documents in the cluster $c_i \in C$; $|l_j|$ be the number of documents in the class $l_j \in L$; $|c_i \cap l_j|$ be the number of documents both in a cluster c_i and a class l_j . Then, the two standard evaluation measures are defined as follows.

Overall F-measure The *F-measure* is often employed to evaluate the accuracy of clustering results. Fung *et al.* [17] measured the quality of a clustering result C using the weighted sum of such maximum *F*-measures for all natural classes according to the cluster size. This measure is called the overall *F*-measure of C , denoted $F(C)$, which is defined as follows.

$$F(C) = \sum_{l_j \in L} \frac{|l_j|}{|D|} \max_{c_i \in C} \{F\}, \text{ where } F = \frac{2PR}{P+R}, P = \frac{|c_i \cap l_j|}{|c_i|} \text{ and } R = \frac{|c_i \cap l_j|}{|l_j|} \quad (3.10)$$

In general, the higher the $F(C)$ values, the better the clustering solution is.

Improvement Ratio To compute a ratio signifying how much improvement is achieved for our proposed approach, F^2IHC , when compared to FIHC method. The *Improvement Ratio (IR)* is the relative value of improvements to the $F(C)$ value of F^2IHC . In the following, we defined the *IR*:

$$IR = \frac{F(C)^{F^2IHC} - F(C)^{FIHC}}{F(C)^{FIHC}} \quad (3.11)$$

where $F(C)^{F^2IHC}$ and $F(C)^{FIHC}$ represent the $F(C)$ values of F^2IHC and FIHC, respectively. A higher IR value indicates that the clustering quality of F^2IHC method is better than the clustering quality of FIHC.

3.4.3 The Effect of Feature Selection

In document clustering, feature selection is essential to make the clustering task efficient and more accurate. The most important goal of feature selection is to extract topic-related terms, which could present the content of each document.

Before applying F^2IHC , we first consider the feature selection strategy. To select the most representative features, we use Formulas (3.1), (3.2), and (3.3) to obtain three weights and select these terms, which their weights are all higher than the pre-defended thresholds. Table 3-14 shows the keyword statistics of our test datasets and the suggested thresholds for each dataset.

Table 3-14: Keyword statistics of our test datasets.

| Datasets | Number of Terms | Number of Key Terms | Percentage of Term Removed | Parameter (α threshold) | Parameter (β threshold) | Parameter (γ threshold) |
|----------|-----------------|---------------------|----------------------------|---------------------------------|--------------------------------|---------------------------------|
| Classic | 41681 | 41251 | 1.0% | 0.028 | 0.01 | 0.005 |
| Hitech | 126737 | 20830 | 83.6% | 0.015 | 0.04 | 0.0008 |
| Re0 | 2886 | 2696 | 6.6% | 0.01 | 0.05 | 0.0005 |
| Reuters | 16641 | 14679 | 11.8% | 0.05 | 0.06 | 0.003 |
| Wap | 8460 | 8021 | 5.2% | 0.01 | 0.07 | 0.0007 |

3.4.4 Experimental Results and Analysis

We have conducted experiments to compare the accuracy of our algorithm F^2IHC with other methods in Section 3.3.4.1 In Section 3.3.4.2, we further evaluate the accuracy of F^2IHC with respect to different MinSup parameters ranging from 2% to 9%. The efficiency of our algorithm is measured in Section 3.3.4.3.

3.4.4.1 Accuracy Comparison

Table 3-15 presents the obtained overall F -Measure values for F^2IHC and $FIHC$ algorithms by comparing four different numbers of clusters, namely 3, 15, 30, and 60. We use the same minimum support, ranging from 3% to 6%, to test $FIHC$ and F^2IHC in each data set, and list their average overall F -Measure values.

It is apparent that the average accuracy of F^2IHC is superior to that of all other algorithms. Although the performances of UPGMA, Bisecting k -means, and $FIHC$ are slightly better than that of F^2IHC in several cases, we argue that the exact number of clusters in a document set is usually unknown in real case, and F^2IHC is robust enough to produce stable, consistent and high quality clusters for a wide range number of clusters. This can be realized by observing the average overall F -measure

values of all test cases. Notice that as UPGMA is not available for large data sets because some experimental results cannot be generated for UPGMA, and we denoted them as N.A.

Table 3-15: Comparison of the overall F-Measure.

| Datasets (# of Natural Clusters) | # of Clusters | F²IHC | FIHC | UPGMA | Bi. <i>k</i>-means |
|---|--------------------------|-------------------------|-------------|--------------|-------------------------------|
| Classic(4) | 3 | 0.51 | 0.53 | N.A. | 0.59 * |
| | 15 | 0.53 * | 0.53 * | N.A. | 0.46 |
| | 30 | 0.54 * | 0.52 | N.A. | 0.43 |
| | 60 | 0.56 * | 0.45 | N.A. | 0.27 |
| | Average | 0.54 * | 0.51 | N.A. | 0.44 |
| Hitech (6) | 3 | 0.47 | 0.48 | 0.33 | 0.54 * |
| | 15 | 0.47 * | 0.45 | 0.33 | 0.44 |
| | 30 | 0.48 * | 0.46 | 0.47 | 0.29 |
| | 60 | 0.45 * | 0.42 | 0.40 | 0.21 |
| | Average | 0.47 * | 0.45 | 0.38 | 0.37 |
| Re0 (13) | 3 | 0.55 * | 0.40 | 0.36 | 0.34 |
| | 15 | 0.54 * | 0.41 | 0.47 | 0.38 |
| | 30 | 0.54 * | 0.38 | 0.42 | 0.38 |
| | 60 | 0.54 * | 0.40 | 0.34 | 0.28 |
| | Average | 0.54 * | 0.40 | 0.40 | 0.34 |
| Reuters (65) | 3 | 0.49 * | 0.48 | N.A. | 0.48 |
| | 15 | 0.56 * | 0.47 | N.A. | 0.42 |
| | 30 | 0.57 * | 0.47 | N.A. | 0.35 |
| | 60 | 0.54 * | 0.42 | N.A. | 0.30 |
| | Average | 0.54 * | 0.46 | N.A. | 0.39 |
| Wap (20) | 3 | 0.39 | 0.37 | 0.39 | 0.40 * |
| | 15 | 0.61 * | 0.49 | 0.49 | 0.57 |
| | 30 | 0.62 * | 0.56 | 0.58 | 0.44 |
| | 60 | 0.62 * | 0.59 | 0.59 | 0.37 |
| | Average | 0.56 * | 0.50 | 0.51 | 0.45 |

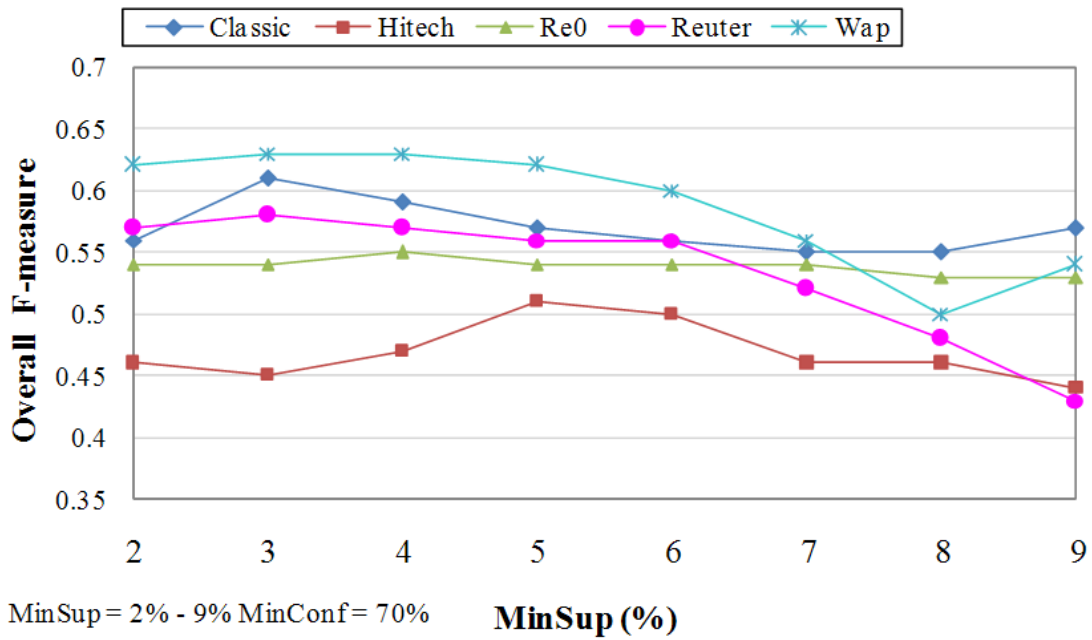
N.A. means not available for large datasets * means the best competitor
The experimental results of UPGMA and Bi. *k*-means are the same as that of FIHC.

From the experimental result in Table 3-15, based on Formula (3.11), our proposed approach has gained $(0.54-0.4)/0.4 = 35\%$ and $(0.54-0.42)/0.42 = 28\%$ $F(C)$ value improvement in average on Re0 and Reuters datasets, respectively, compared with FIHC algorithm. For the other datasets, the reasons for limited improvement may be due to the numbers of clusters were fixed with 3, 15, 30, and 60 for comparison purpose, and these numbers of clusters may not be appropriate for these datasets.

3.4.4.2. Sensitivity to Various Parameters

Our algorithm has two main parameters for the adjustment of accuracy quality. We now discuss how the default values were chosen, the effects of modifying those parameters, and suggestions for practical uses. The first one is mandatory and is denoted MinSup, which means the minimum support for fuzzy frequent itemsets generation. The other is optional, and is denoted KCluster, which represents the number of clusters at level 1 of the cluster tree. In Table 3-15, we do not only demonstrate the accuracy of the produced solutions, but also show the sensitivity of the accuracy of KCluster.

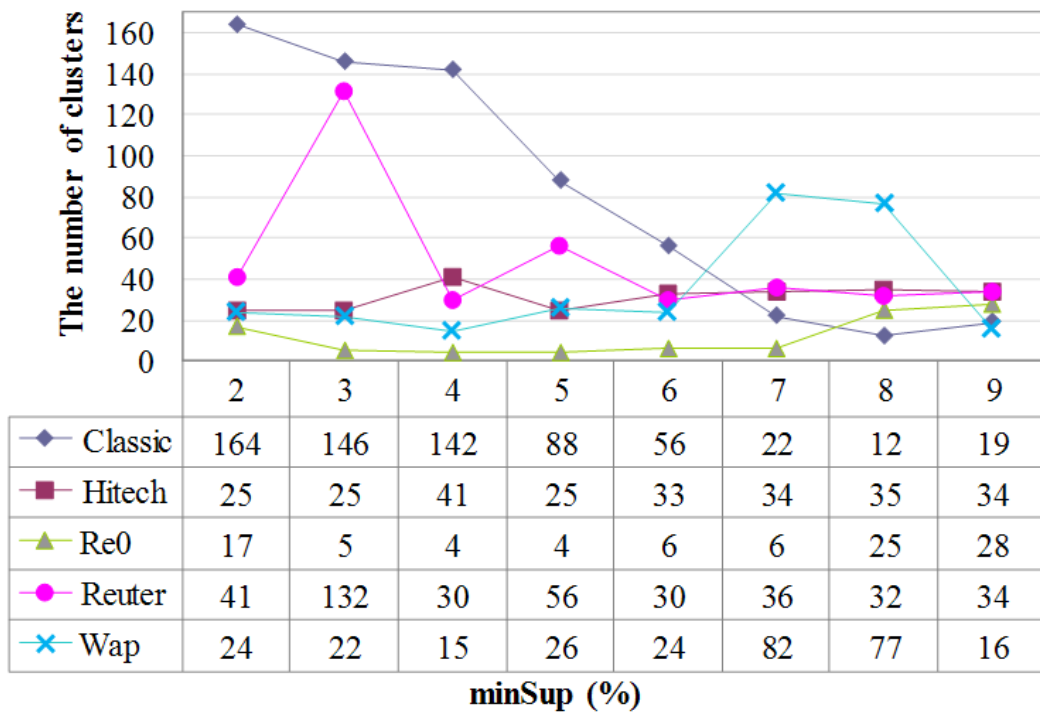
Figure 3-10(a) depicts the overall F -measure values of F^2IHC when accepting different mandatory parameters, but ignoring the parameter values of the optional ones. We observe that high clustering accuracies are fairly consistent while MinSup are set between 2% and 9%. As KClusters is not specified in each test case, the sibling merging algorithm has to decide the most appropriate number of output clusters, which are shown in Figure 3-10(b).



MinSup = 2% - 9% MinConf = 70%

MinSup (%)

(a)



minSup (%)

(b)

Figure 3-10: The accuracy test of F^2IHC for different MinSup values with the optimal cluster numbers determined by the sibling merging algorithm.

Based on our test, we observe a general guidance that the best choice of MinSup can be set between 3% and 6%. Nevertheless, it cannot be over emphasized that MinSup should not be regarded as the only parameter for finding the optimal

accuracy. It is supposed that users are responsible to adjust the shape of the cluster tree based on the value of MinSup. The smaller the value of MinSup, the deeper (and broader) a cluster tree can be generated, and vice versa.

3.4.4.3. Efficiency and Scalability

Our algorithm involves three major phases: finding fuzzy frequent itemsets, initial clustering, and clusters merging. Figure 3-11 depicts the average execution time of F²IHC algorithm on five datasets, where there were five different MinSup, 5% ~ 9%, set to evaluate the performance. According to the result shown in Figure 3-11, the document length dominates the performance of the execution time. From Figure 3-11, we further found that the average execution time of the fuzzy mining stage on five datasets is almost identical. The runtime of our algorithm is inversely related to the input parameter MinSup. In other words, runtime increases as MinSup decreases. Due to the longer average length of documents in Hitech and Wap datasets, their average initial clustering and cluster merging time is higher than that of the other datasets.

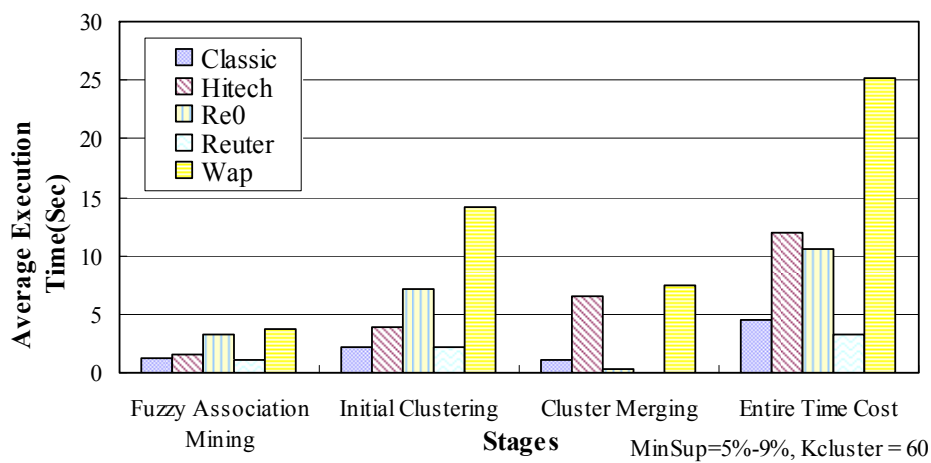


Figure 3-11: The detailed time cost analysis of F²IHC on five datasets.

To analyze the scalability of our algorithm, we get 100,000 documents from RVC1 (Reuters Corpus Volume 1) dataset [31], which contains news from Reuters Ltd. There are three category sets: Topics, Industries, and regions. In our experiments, we consider the Topics category set, which includes 23,149 training and 781,265 testing documents. Before clustering this dataset, documents were parsed by converting all terms in documents into lower case, removing stop words, and applying the stemming algorithm.

Figure 3-12 shows the runtimes with respect to the different sizes of RVC1 dataset, ranging from 10K to 100K documents, for different stages of our algorithm. The whole process was completed within five minutes. The figure also shows that fuzzy mining and the initial clustering stages are the most two time-consuming stages in our algorithm. In the clustering stage, most of the time is spent on constructing initial clusters and its runtime is almost linear with respect to the number of documents.

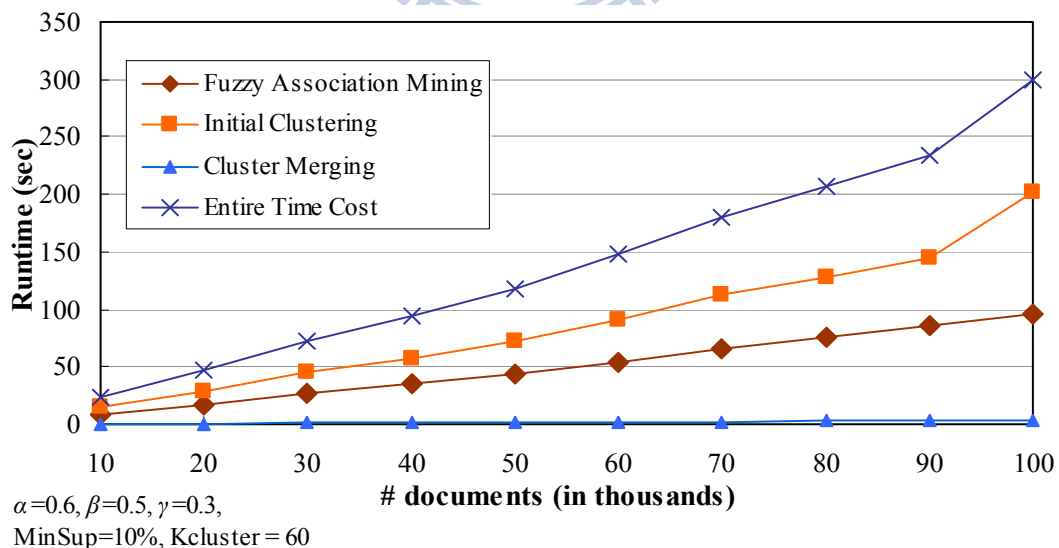


Figure 3-12: Scalability of F²IHC.

3.5 Summary

Although numerous interesting document clustering methods have been extensively studied for many years, the high computation complexity and space need still make the clustering methods inefficient. Hence, reducing the heavy computational load and increasing the precision of the unsupervised clustering of documents are important issues. In this chapter, we derived a frequent itemset-based hierarchical document clustering approach, based on the fuzzy association rule mining, for alleviating these problems satisfactorily. In our approach, we start with the document pre-processing stage; then employ the fuzzy association data mining method in second stage; automatically generate a candidate cluster set, and merge the high similar clusters, and finally build a hierarchical cluster tree in a top-down fashion for easy browsing. Our experiments show that the accuracy of our algorithm is higher than that of FIHC method, UPGMA, and Bisecting k -means when compared on the five standard datasets. Moreover, the experiment results show that the use of fuzzy association rule mining discovery important candidate clusters for document clustering to increase the accuracy quality of document clustering. Therefore, it is worthy extending in reality for concentrating on huge text documents management.

Chapter 4

Fuzzy Frequent Itemset-based Document Clustering (F²IDC) Approach

Many documents contain the similar semantic information, even though they do not contain common words. For instance, if one document describes the ‘apple’ issue, it should be turned up ‘fruit’ issue even though the document does not contain term ‘fruit’. In order to consider the conceptual similarity of terms that do not co-occur actually, we employ WordNet in our document clustering approach and show where and how it can be fruitfully utilized.

However, most frequent itemset-based clustering algorithms only account for term frequency in the documents and all ignore the important semantic relationships between terms. Therefore, our approach aims to investigate whether or not WordNet semantic relationships can improve the clustering quality of frequent itemset-based clustering

In this chapter, we propose an effective Fuzzy Frequent Itemset-based Document Clustering (F²IDC) approach based on fuzzy association rule mining in conjunction with WordNet for clustering textual documents. In contrast to F²IHC approach proposed in Chapter 3, this chapter illustrates how to add hypernyms as term features for the document representation, how to utilize the hypernyms in the process of fuzzy association rule mining to obtain the conceptual labels from the derived clusters.

The overall process and detail design of the proposed F²IDC approach is shown in Figure 4-1. The process of F²IDC is similar to the general process of document clustering (as depicted in Figure 2-1), expect for the gray-colored components (i.e.,

Document Enrichment, Fuzzy Frequent Itemset Mining, and Clustering, etc.) In the following, we explain these three stages in our framework:

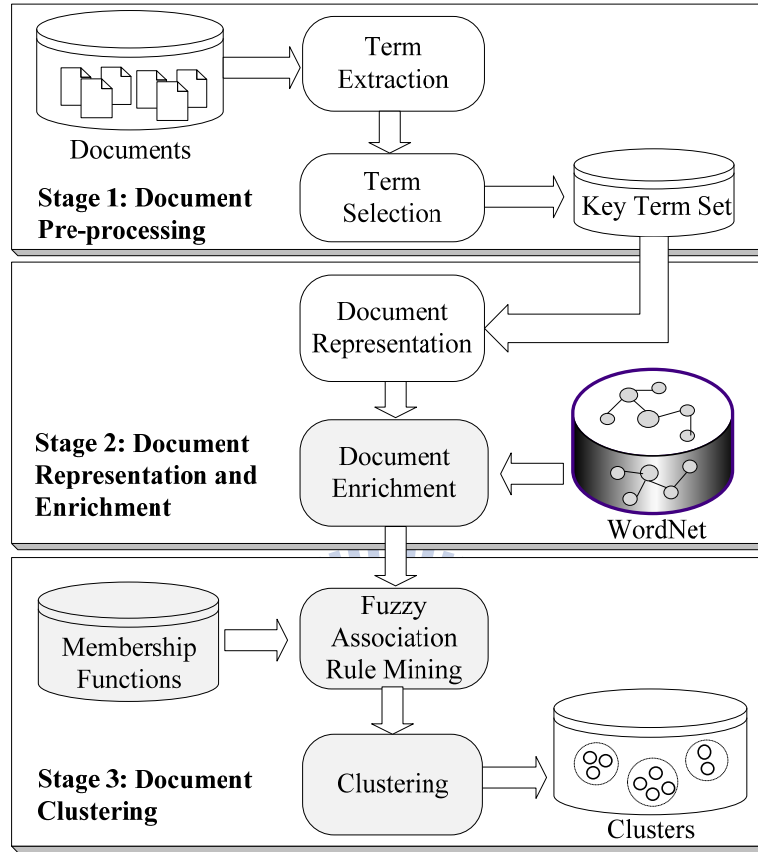


Figure 4-1: The F²IDC framework.

4.1 Stage 1: Document Analyzing

As with document clustering techniques, the proposed approach starts with term extraction. For a document set $D = \{d_1, d_2, \dots, d_i, \dots, d_n\}$, a term set $T_D = \{t_1, t_2, \dots, t_j, \dots, t_s\}$, which is the set of terms appeared in D , can be obtained. The details of the term extraction are described in Section 2.1.

The feature description of a document is constituted by terms of the document set to form a term vector. A term vector with high dimensions is easy to make clustering inefficient and difficult in principle. Hence, we employ tf-idf [46] as the

feature selection method to produce a low dimensional term vector. A term will be discarded if its weight is less than a tf-idf threshold γ . Formula (4.1) is used for the measurement of $tfidf_{ij}$ for the importance of a term t_j within a document d_i . For preventing a bias for longer documents, the weighted frequency of each term is usually normalized by the maximum frequency of all terms in d_i , and is defined as follows:

$$tfidf_{ij} = 0.5 + 0.5 * \frac{f_{ij}}{\max_{t_j \in d_i}(f_{ij})} \times \log\left(1 + \frac{|D|}{|\{d_i | t_j \in d_i, d_i \in D\}|}\right) \quad (4.1)$$

where f_{ij} is the frequency of t_j in d_i , and the denominator is the maximum frequency of all terms in d_i . $|D|$ is the total number of documents in the document set D , and $|\{d_i | t_j \in d_i, d_i \in D\}|$ is the number of documents containing t_j .

After the step of term selection, the *key term set* of D , denoted $K_D = \{t_1, t_2, \dots, t_j, \dots, t_p\}$ is obtained. K_D is a subset of T_D , including only meaningful key terms, and satisfying the pre-defined minimum tf-idf threshold γ .

4.2 Stage 2: Document Representation and Enrichment

In this stage, each document d_i in D is represented using those terms in K_D . Thus, each document $d_i \in D$, denoted $d_i = \{(t_1, f_{i1}), (t_2, f_{i2}), \dots, (t_j, f_{ij}), \dots, (t_m, f_{im})\}$, is represented by a set of pairs (term, frequency), where the frequency f_{ij} represents the occurrence of the key term t_j in d_i .

Accordingly, we enrich the document representation by using WordNet, a source repository of semantic meanings. WordNet, developed by Miller *et al.* [40], consists of so-called synsets, together with a hypernym/hyponym hierarchy.

The basic idea of document enrichment is to add the generality of terms by corresponding hypernyms of WordNet based on the key terms appeared in each document. Each key term is linked up to the top 5 levels of hypernyms. After key terms are extracted from the document set, they can be organized based on the hierarchical (IS-A) relationship of WordNet [40] to construct term trees. A term tree (defined in Definition 4.1) is constructed by matching a key term in WordNet and then navigating upwards for the top 5 levels of hypernyms. Eventually, all term trees can be regarded as a term forest (defined in Definition 4.2) for the document set D .

Definition 4.1: A *term tree* of term t , denoted $\mathcal{J} = (W, H, I, t)$, is a 4-tuple consisting of a set of hypernyms $I = \{h_1, \dots, h_r\}$ of a key term $t_j \in W$, together with their reference function $H: 2^W \mapsto 2^I$ in W , where W represents the WordNet and H links the set of hypernyms up to five levels in W . We denote $h_1 \leq h_2$, when h_2 is the hypernym of h_1 defined in W .

Definition 4.2: A *term forest* of a set of terms $\{t_1, t_2, \dots, t_i, \dots, t_m\}$, denoted $\mathcal{F} = \{\mathcal{J}_1, \mathcal{J}_2, \dots, \mathcal{J}_i, \dots, \mathcal{J}_m\}$, is a set of term trees, where m is the total number of key terms in D .

Using hypernyms can help our approach magnify hidden similarities to identify related topics, which potentially leads to better clustering quality [24][49]. Hence, we enriched the representation of each document with hypernyms based on WordNet to find semantically-related documents. Based on the key terms appeared in a document, the representation of this document is enriched by associating them with the term trees accordingly. For a simple and effective combination, these added hypernyms form a new key term set, denoted $K_D = \{t_1, t_2, \dots, t_m, h_1, \dots, h_r\}$, where h_j is a

hypernym. The enriched document d_i is represented by $d_i = \{(t_1, f_{i1}), (t_2, f_{i2}), \dots, (t_p, f_{ip}), (h_1, hf_{i1}), \dots, (h_r, hf_{ir})\}$, where a weight of 0 will be assigned to several terms appearing in some of the documents but not in d_i . The frequency f_{ij} of a key term t_j in d_i is mapped to its hypernyms $\{h_1, \dots, h_j, \dots, h_r\}$ to accumulate as the frequency hf_{ij} of h_j .

The reason of using hypernyms of WordNet is that hypernyms can reveal hidden similarities to identify related topic, potential leading to the better clustering quality [49]. For example, a document about ‘sale’ may not be associated to a document about ‘trade’ by the clustering algorithm if there are only ‘sale’ and ‘trade’ in the key term set. But, if the more general term ‘commerce’ is added to both documents, their semantic relation is revealed. The suitable representation of each document for the later mining can be derived by Algorithm 4.1 shown in Figure 4-2.

Algorithm 4.1. Basic algorithm to obtain the designated representation of all documents

Input: A document set D ; A well-defined stop word list; WordNet W ; The minimum tf-idf threshold γ .

Output: The formal representation of all documents in D .

1. Extract the term set $T_D = \{t_1, t_2, \dots, t_j, \dots, t_s\}$
 2. Remove all stop words from T_D
 3. Apply Stemming for T_D
 4. For each $d_i \in D$ do //key term selection
 - For each $t_j \in T_D$ do
 - (1) Evaluate its $tfidf_{ij}$ weight // defined by Formula (4.1) in Section 4
 - (2) Retain the term if $tfidf_{ij} \geq \gamma$
 5. Form the key term set $K_D = \{t_1, t_2, \dots, t_j, \dots, t_m\}$, where $m \leq s$
 6. For each $t_j \in K_D$ do //refer to W
 - $= (W, I, H, t_j)$ // find the set of hypernyms $I = \{h_1, \dots, h_r\}$ and their links H
 7. Form the Term Forest $\mathcal{F} = \{\mathcal{J}_1, \mathcal{J}_2, \dots, \mathcal{J}_i, \dots, \mathcal{J}_m\}$
 8. For each $d_i \in D$ do //document enrichment step
 - For each $t_j \in K_D$ do
 - (1) If (h_j is hypernyms of t_j) then //refer to W
 - (a) $hf_{ij} \rightarrow hf_{ij} + f_{ij}$
 - (2) If (h_j is not in K_D) then
 - (b) $K_D \rightarrow K_D \cup \{h_j\}$
 9. For each $d_i \in D$ do //in order to decrease noise from hypernyms, tf-idf method is executed again
 - For each $t_j \in K_D$ do
 - (1) Evaluate its $tfidf_{ij}$ weight
 - (2) Retain the term if $tfidf_{ij} \geq \gamma$
 10. Form the new key term set $K_D = \{t_1, t_2, \dots, t_m, h_1, \dots, h_r\}$
 11. For each $d_i \in D$, record the frequency f_{ij} of t_j and the frequency hf_{ij} of h_j in d_i to obtain the final representation of $d_i = \{(t_1, f_{i1}), (t_2, f_{i2}), \dots, (t_m, f_{im}), (h_1, hf_{i1}), \dots, (h_r, hf_{ir})\}$
-

Figure 4-2: The detailed description of Algorithm 4.1.

4.3 Stage 3: Document Clustering

The final stage is to group the documents into clusters. In the following, we first define the membership functions and present our fuzzy association rule mining algorithm for texts. Subsequently, based on the mining results, we illustrate the details of the clustering process.

4.3.1 The Fuzzy Association Rule Mining Algorithm for Texts

According to Definition 3.5, the corresponding membership functions $w_{ij}^r(f_{ij})$ are defined by Formulas (4.2), (4.3), and (4.4), respectively. The derived membership functions are shown in Figure 4-3.

$$w_{ij}^{Low}(f_{ij}) = \begin{cases} 0, & f_{ij} = 0 \\ 1 + \frac{f_{ij} - a}{b - a}, & a \leq f_{ij} \leq b \\ 2, & b < f_{ij} < c \\ 1 + \frac{f_{ij} - d}{c - d}, & c \leq f_{ij} \leq d \\ 1, & f_{ij} > d \end{cases}, \quad \begin{matrix} a = 0, \\ b = \min(f_{ij}), \\ c = \frac{\lceil \text{avg}(f_{ij}) + \min(f_{ij}) \rceil}{2}, \\ d = \text{avg}(f_{ij}) \end{matrix} \quad (4.2)$$

$$w_{ij}^{mid}(f_{ij}) = \begin{cases} 0, & f_{ij} = 0 \\ 1, & f_{ij} < a \\ 1 + \frac{f_{ij} - a}{b - a}, & a \leq f_{ij} \leq b \\ 2, & b < f_{ij} < c \\ 1 + \frac{f_{ij} - d}{c - d}, & c \leq f_{ij} \leq d \\ 1, & f_{ij} > d \end{cases}, \quad \begin{matrix} a = \min(f_{ij}), \\ b = \frac{\lceil \text{avg}(f_{ij}) + \min(f_{ij}) \rceil}{2}, \\ c = \text{avg}(f_{ij}), \\ d = \text{avg}(f_{ij}) + \frac{\lceil \max(f_{ij}) - \text{avg}(f_{ij}) \rceil}{4} \end{matrix} \quad (4.3)$$

$$w_{ij}^{high}(f_{ij}) = \begin{cases} 0, & f_{ij} = 0 \\ 1, & f_{ij} < a \\ 1 + \frac{f_{ij} - a}{b - a}, & a \leq f_{ij} \leq b \\ 2, & b < f_{ij} < c \\ 1 + \frac{f_{ij} - d}{c - d}, & c \leq f_{ij} \leq d \end{cases}, \quad \begin{matrix} a = \text{avg}(f_{ij}), \\ b = \text{avg}(f_{ij}) + \frac{\lceil \max(f_{ij}) - \text{avg}(f_{ij}) \rceil}{4}, \\ c = \text{avg}(f_{ij}) + \frac{\lceil \max(f_{ij}) - \text{avg}(f_{ij}) \rceil}{2}, \\ d = \max(f_{ij}) \end{matrix} \quad (4.4)$$

In Formulas (4.2), (4.3), and (4.4), $\min(f_{ij})$ is the minimum frequency of terms in D , $\max(f_{ij})$ is the maximum frequency of terms in D , and $\text{avg}(f_{ij}) = \lceil \frac{\sum_{i=1}^n f_{ij}}{|K|} \rceil$, where $f_{ij} \neq \min(f_{ij})$ or $\max(f_{ij})$, and $|K|$ is the number of summed key terms.

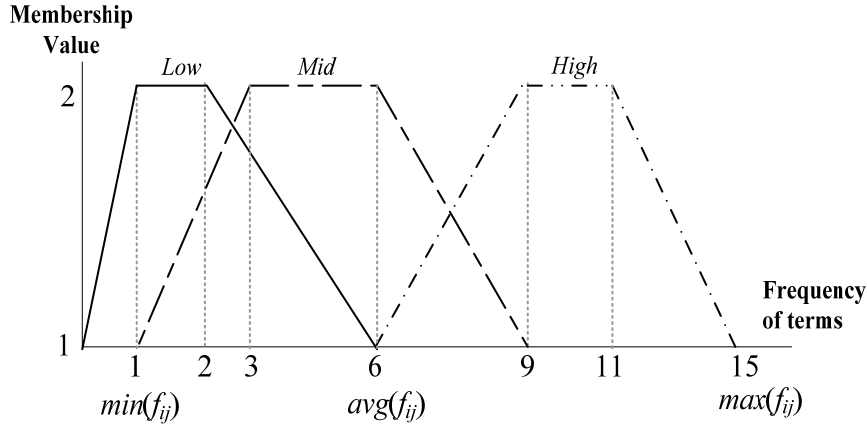


Figure 4-3: The predefined membership functions.

Then, we use the membership functions shown in Figure 4-3 and Algorithm 3.2 (shown in Figure 3-4) to generate the candidate cluster set as output.

4.3.2 Clustering

The objective of Algorithm 4.2 shown in Figure 4-4 is to assign each document to the best fitting cluster c_i^q , and finally obtain the target cluster set for output. The assignment process is based on the Document-Cluster Matrix (DCM) derived from the Document-Term Matrix (DTM) and the Term-Cluster Matrix (TCM). We define DTM and TCM by Definitions 3.8 and 3.9, respectively. The DCM of a document set D is defined by Definition 3.10. For avoiding low the clustering accuracy, the inter-cluster similarity (defined by Formula (3.9) in Chapter 3) between two target clusters is calculated to merge the small target clusters with the similar topic.

Algorithm 4.2: Basic algorithm to obtain the target clusters

Input: A document set $D = \{d_1, d_2, \dots, d_i, \dots, d_n\}$; The key term set $K_D = \{t_1, t_2, \dots, t_j, \dots, t_m\}$; The candidate cluster set $\tilde{C}_D = \{\tilde{c}_1^1, \dots, \tilde{c}_{l-1}^1, \tilde{c}_l^q, \dots, \tilde{c}_k^q\}$; A minimum *Inter-Sim* threshold δ ;

Output: The target cluster set $C_D = \{c_1^1, c_2^1, \dots, c_i^q, \dots, c_f^q\}$

1. Build $n \times m$ document-term matrix $W = [w_{ij}^{\max-R_j}]$. // $w_{ij}^{\max-R_j}$ is the weight (fuzzy value) of t_j in d_i and $t_j \in L_1$.

2. Build $m \times k$ term-cluster matrix $G = [g_{jl}^{\max-R_j}]$. // $g_{jl}^{\max-R_j} = \frac{\text{score}(\tilde{c}_l^q)}{\sum_{i=1}^n w_{ij}^{\max-R_j}}$, $1 \leq j \leq m$, $1 \leq l \leq k$, and

$$\text{score}(\tilde{c}_l^q) = \sum_{d_i \in \tilde{c}_l^q, t_j \in \tau} w_{ij}^{\max-R_j}, \text{ where } w_{ij}^{\max-R_j} \text{ is the weight (fuzzy value) of } t_j \text{ in } d_i \in \tilde{c}_l^q \text{ and } t_j \in L_1.$$

3. Build $n \times k$ document-cluster matrix $V = W \cdot G = [v_{il}] = \sum_{m=1}^m w_{im} g_{ml}$.

4. Based on V , assign d_i to a target cluster c_l^q

(1) $c_l^q = \{d_i \mid v_{il} = \max\{v_{i1}, v_{i2}, \dots, v_{ik}\} \in \tilde{c}_l^q, \text{ where the number of } v_{il} \text{ is } 1\}$, otherwise (2).

(2) $c_l^q = \{d_i \mid v_{il} = \max\{v_{i1}, v_{i2}, \dots, v_{ik}\} \in \tilde{c}_l^q, \text{ where the number of } v_{il} > 1 \text{ and } \tilde{c}_l^q \text{ with the highest fuzzy count value corresponding to its fuzzy frequent itemset}\}$.

5. Clusters merging

(1) For each $c_l^q \in C_D$ do

(a) If ($c_l^q = \text{null}$) then $\{\text{remove this target clusters } c_l^q \text{ from } C_D\}$

(2) For each pair of target clusters $(c_x^q, c_y^q) \in C_D$ do

(a) Calculate the *Inter_sim*

(b) Store the results in the Inter-Cluster Similarity matrix I .

(3) If (one of the *Inter_sim* value in $I \geq \delta$) then

(a) Select (c_x^q, c_y^q) with the highest *Inter_sim*.

(b) Merge the smaller target cluster into the larger target cluster.

(c) Repeat Step (2) to update I

6. Output C_D

Figure 4-4: The detailed description of Algorithm 4.2.

4.4 An Illustrative Example of F²IDC Method

Suppose we have a document set $D = \{d_1, d_2, \dots, d_5\}$ and its key term set

$K_D = \{\text{sale, trade, medical, health}\}$. Figure 4-5 illustrates the process of Algorithm 4.1 to obtain the representation of all documents. This representation scheme will be employed in the following to illustrate our approach.

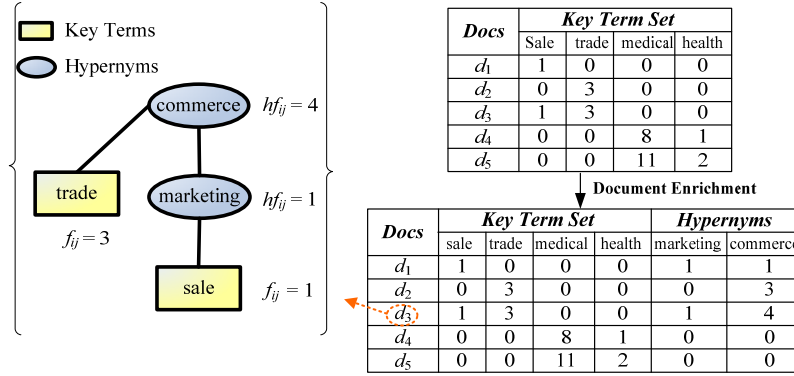


Figure 4-5: The process of Algorithm 4.1 of this example.

Consider the representation of all documents generated by Algorithm 4.1 in Figure 4-5, the membership functions defined in Figure 4-3, the minimum support value 70%, and the minimum confidence value 70% as inputs. The fuzzy frequent itemsets discovery procedure is depicted in Figure 4-6.

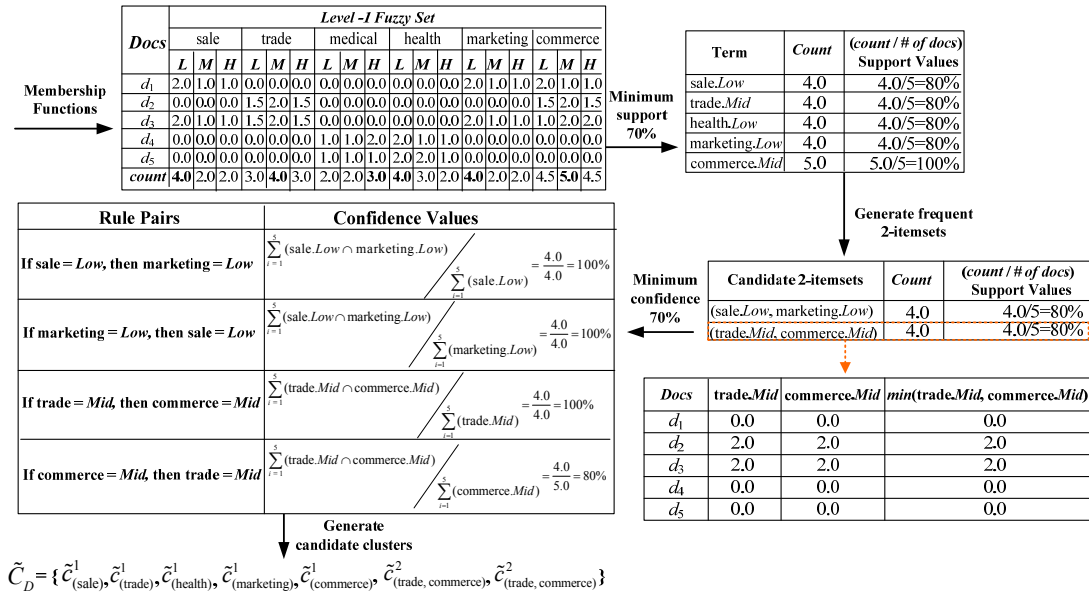


Figure 4-6: The process of Algorithm 3.2 of this example.

Moreover, consider the candidate cluster set \tilde{c}_D was already generated in Figure 4-6. Now, suppose the minimum *Inter-Sim* value is 0.5. Figure 4-7 illustrates the process of Algorithm 4.2 and shows the final results.

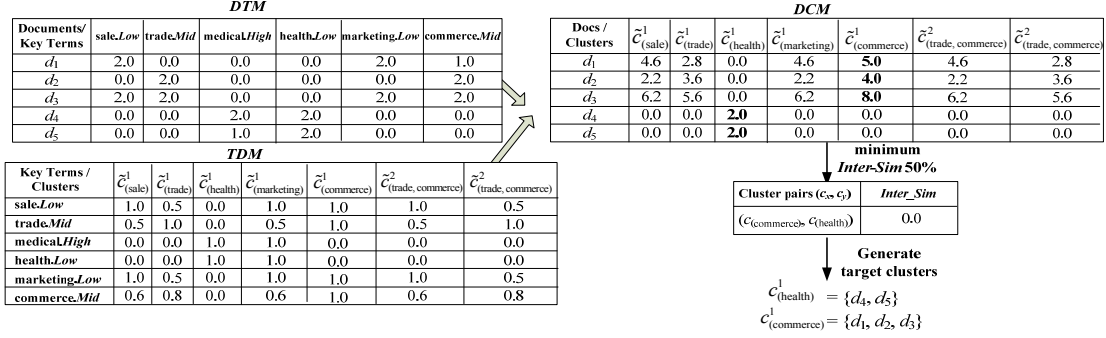


Figure 4-7: The process of Algorithm 4.2 of this example.

4.5 Experiments

In this section, we experimentally evaluated the performance of the proposed algorithm by comparing with that of FIHC, *k*-means, Bisecting *k*-means, and UPGMA algorithms. We make use of the FIHC 1.0 tool to generate the results of FIHC. Moreover, Steinbach *et al.* [53] compared the performance of some influential clustering algorithms, and the results indicated that UPGMA and Bisecting *k*-means are the most accurate clustering algorithms. Therefore, the CLUTO-2.1.2a⁷ Clustering tool is applied to generate the results of *k*-means, Bisecting *k*-means, and UPGMA. The produced results are then fetched into the same evaluation program to ensure a fair comparison. All the experiments were performed on a P4 3.2GHz Windows XP machine with 1GB memory. The implementation was written with Java 1.5 to allow reusability of the written code.

⁷ <http://glaros.dtc.umn.edu/gkhome/views/cluto/>

4.5.1 Datasets

To test the proposed approach, we used four different kinds of datasets: *Classic*, *Re0*, *R8*, and *WebKB*, which are widely adopted as standard benchmarks for the text categorization task. They are heterogeneous in terms of document size, cluster size, number of classes, and document distribution. Moreover, these datasets are not specially designed to combine with WordNet for facilitating the clustering result.

Table 4-1 summarizes the statistics of these datasets. Each document is pre-classified into a single topic, i.e., a natural class. The class information is utilized in the evaluation method for measuring the accuracy of the clustering result. The detailed information of these datasets is described as follows:

- *Classic*⁸: This dataset is a combination of the four classes CACM, CISI, CRAN, and MED abstracts. *Classic* includes 3,204 CACM documents, 1,460 CISI documents from information retrieval papers, 1,398 CRANFIELD documents from aeronautical system papers, and 1,033 MEDLINE documents from medical journals.
- *Re0*⁹: *Re0* is a dataset, derived from Reuters-21578¹⁰ text categorization test collection Distribution 1.0. *Re0* includes 1,504 documents belonging to 13 different classes.
- *R8*¹¹: *R8* is a subset of the Reuters-21578 text categorization collections. It considers only the documents associated with a single topic and the classes which

⁸ <ftp://ftp.cs.cornell.edu/pub/smart/>

⁹ The preprocessed datasets can be downloaded. <http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download/>

¹⁰ <http://www.daviddlewis.com/resources/testcollections/>

¹¹ The preprocessed datasets can be downloaded. <http://web.ist.utl.pt/~acardoso/datasets/>

still have at least one train and one test example. *R8* includes 7,674 documents with 8 most frequent classes.

- *WebKB*¹²: This dataset consists of web pages collected by the WebKB project of the CMU text learning group [10]. These pages are manually classified into seven classes. In our test, we select the four most popular entity-representing classes: course, faculty, project, and student.

Table 4-1: Statistics for our test datasets.

| Datasets | Documents | Classes | Class Size | | | Document Length |
|----------|-----------|---------|------------|---------|------|-----------------|
| | Total | Total | Max | Average | Min | Average |
| Classic | 7,094 | 4 | 3203 | 1774 | 1033 | 43 |
| Re0 | 1,504 | 13 | 608 | 116 | 11 | 76 |
| R8 | 7,674 | 8 | 3,923 | 959 | 51 | 48 |
| WebKB | 4,199 | 4 | 1,641 | 1050 | 504 | 124 |

4.5.2 Parameters Selection

Table 4-2 summarizes the parameters for our proposed method and the other algorithms to compare the clustering performance.

Before applying F^2IDC , we first consider the feature selection strategy. In order to select the most representative features, we use Formula (4.1) to obtain the key terms with weights higher than the pre-defined thresholds γ . Table 4-3 shows the keyword statistics of our test datasets and the suggested threshold for each dataset.

¹² The preprocessed datasets can be downloaded. <http://www.cs.technion.ac.il/~ronb/thesis.html>

Table 4-2: List of all parameters for our algorithms and the other three algorithms.

| Parameter Name | F ² IDC | FIHC | UPGMA ^{13,14} | Bi. k-means ¹⁵ |
|-----------------------------|---------------------------|--------|------------------------|---------------------------|
| Datasets | Classic, Re0, R8, WebKB | | | |
| Stopword Removal | Yes | | | |
| Stemming | Yes | | | |
| Length of the smallest term | Three | | | |
| Weight of the term vector | tf | tf-idf | tf-idf | tf-idf |
| Levels of hypernyms | <i>H1, H2, H3, H4, H5</i> | | | |
| Cluster count <i>k</i> | 3, 15, 30, 60 | | | |

H1 represents the addition of direct hypernyms; *H2* stands for the addition of hypernyms of the first and second levels, and so on.

Table 4-3: Keyword statistics of our test datasets.

| Datasets | # of Terms | # of Terms after pre-processing | # of Terms after Enriching | γ threshold | |
|----------|------------|---------------------------------|----------------------------|--------------------|----------------------------------|
| | | | | F ² IDC | WordNet-based F ² IDC |
| Classic | 40,291 | 40,279 | 41,931 | 0.60 | 0.65 |
| Re0 | 2,886 | 2,678 | 3,507 | 0.60 | 0.65 |
| R8 | 16,810 | 16,790 | 18,692 | 0.60 | 0.65 |
| WebKB | 42,503 | 34,310 | 36,622 | 0.60 | 0.65 |

The two algorithms, F²IDC and FIHC, both have two main parameters for the adjustment of accuracy quality. This first one is mandatory and is denoted MinSup, which means the minimum support for frequent itemsets generation. The other one is optional, and is denoted KCluster, which represents the number of clusters. As Bisecting *k*-means and UPGMA require a predefined number of clusters as their inputs, their KCluster parameters must be provided.

¹³ The command was `vcluster -clmethod=aggllo -crfun=upgma -sim=cos -rowmodel=maxtf -colmodel=idf -clabelfile=<X>.mat.clabel <X>.mat <K>`.

¹⁴ <X> is the name of the dataset being tested (ex. R8, WebKB etc.), and <K> is the number of clusters desired in the final solution. Vcluster is the name of the Cluto clustering program that clusters data from .mat files as input.

¹⁵ The command was `vcluster -clmethod=rbr -crfun=i2 -sim=cos -cstype=best -rowmodel=maxtf -colmodel=idf -clabelfile=<X>.mat.clabel <X>.mat <K>`.

4.5.3 Experimental Results and Analysis

The experiments were conducted by the following steps. First, we evaluated our method, F²IDC, on the four datasets mentioned above and compared its accuracy with that of FIHC, Bisecting k -means, and UPGMA. Moreover, we verified if the use of WordNet can generate conceptual labels for derived clusters. Second, the dataset, RVC1 (Reuters Corpus Volume 1) [31], was chosen to evaluate the efficiency and scalability of F²IDC.

4.5.3.1. Accuracy Comparison for F²IDC Algorithm

Table 4-4 presents the obtained overall F-Measure values for WordNet-based F²IDC and the other WordNet-based algorithms by comparing four different numbers of clusters, namely 3, 15, 30, and 60, on four datasets respectively. For each algorithm, we run each dataset enriched with the top 5 levels of hypernyms. We tested each algorithm's clustering results with the value H , the levels of hypernyms, from 1 to 5 and selected the best results. We chose the minimum support in {25%, 28%, 30%, 32%, 35%} to run F²IDC with WordNet for all datasets. Moreover, we set the minimum support values, ranging from 3% to 6%, to obtain the best results for FIHC.

It is apparent that the average accuracy of Bisecting k -means and FIHC are slightly better than that of F²IDC in several cases. We argue that the exact number of clusters in a document set is usually unknown in real case, and F²IDC is robust enough to produce stable, consistent and high quality clusters for a wide range number of clusters. This can be realized by observing the average overall F -measure

values of all test cases. Notice that UPGMA is not available for large data sets because some experimental results cannot be generated for UPGMA, and we denoted them as N.A. Since FIHC is not available for the documents of long average length, there is no experimental result generated on the WebKB dataset, and we also marked them as N.A.

Table 4-4: Average overall F-measure comparison for four clustering algorithms on the four datasets.

| Datasets (# of Natural Classes) | # of Clusters | F ² IDC(H) | FIHC(H) | UPGMA(H) | Bi. k-means(H) |
|---------------------------------------|---------------|-----------------------|-----------|----------|----------------|
| Classic (4) | 3 | 0.68(3) * | 0.51(1) | N.A. | 0.61(5) |
| | 15 | 0.70(3) * | 0.51(1) | N.A. | 0.59(5) |
| | 30 | 0.70(3) * | 0.52(1) | N.A. | 0.43(5) |
| | 60 | 0.69(3) * | 0.51(1) | N.A. | 0.28(5) |
| | Average | 0.69(3) * | 0.51(1) | N.A. | 0.48(5) |
| Re0 (13) | 3 | 0.56(3) * | 0.43(1) | 0.40(3) | 0.40(3) |
| | 15 | 0.53(3) * | 0.40(1) | 0.35(3) | 0.42(3) |
| | 30 | 0.52(3) * | 0.39(1) | 0.35(3) | 0.36(3) |
| | 60 | 0.52(3) * | 0.34(1) | 0.35(3) | 0.30(3) |
| | Average | 0.53(3) * | 0.39(1) | 0.36(3) | 0.37(3) |
| R8 (8) | 3 | 0.57(3) * | 0.47(1) | N.A. | 0.59(3) * |
| | 15 | 0.44(3) * | 0.43(1) | N.A. | 0.42(3) |
| | 30 | 0.43(3) * | 0.43(1) * | N.A. | 0.36(3) |
| | 60 | 0.44(3) * | 0.43(1) | N.A. | 0.23(3) |
| | Average | 0.47(3) * | 0.44(1) | N.A. | 0.40(3) |
| WebKB (4) | 3 | 0.48(1) * | N.A. | 0.44(1) | 0.33(3) |
| | 15 | 0.49(1) * | N.A. | 0.43(1) | 0.19(3) |
| | 30 | 0.49(1) * | N.A. | 0.42(1) | 0.13(3) |
| | 60 | 0.49(1) * | N.A. | 0.36(1) | 0.07(3) |
| | Average | 0.49(1) * | N.A. | 0.42(1) | 0.18(3) |

N.A. means not available for the datasets * means the best competitor

The *Improvement Ratio (IR)* is the ratio of improvements to the $F(C)$ value of our proposed approach, F²IDC, when compared with the other compared algorithms.

In the following, we define the *IR*:

$$IR = \frac{F(C)^{F^2IDC} - F(C)^{<Y>}}{F(C)^{<Y>}}, \quad (4.5)$$

where $F(C)^{F^2IDC}$ and $F(C)^{<Y>}$ represent the $F(C)$ values of F²IDC and the other three algorithms (e.g., <Y> can be FIHC, UPGMA, or Bi. *k*-means), respectively. A higher IR value indicates that the clustering quality of F²IDC method is better than the clustering quality of the other algorithms.

From the experimental result in Table 4-4, based on Formula (4.5), our proposed approach has gained $F(C)$ value improvement in average (as shown in Table 4-5) for the other three algorithms on four datasets. The percentage of improvement ratio ranges from 7% to 172% based on the increases of the $F(C)$ value.

Table 4-5: Improvement Ratio for other three clustering algorithms on the four datasets.

| Datasets | Clustering Algorithms | | | | Improvement Ratio | | |
|----------|--------------------------------|------------------|-------------------|---------------------|-------------------|-------|---------------------|
| | F ² IDC(<i>w</i>) | FIHC(<i>w</i>) | UPGMA(<i>w</i>) | Bi. <i>k</i> -means | FIHC | UPGMA | Bi. <i>k</i> -means |
| Classic | 0.69(3) | 0.51(1) | N.A. | 0.48(5) | +0.35 | N.A. | +0.43 |
| Re0 | 0.54(3) | 0.39(1) | 0.36(3) | 0.37(3) | +0.39 | +0.50 | +0.46 |
| R8 | 0.47(3) | 0.44(1) | N.A. | 0.40(3) | +0.07 | N.A. | +0.18 |
| WebKB | 0.49(1) | N.A. | 0.42(1) | 0.18(3) | N.A. | +0.17 | +1.72 |

4.5.3.2. The Effect of Enriching the Document Representation

As described in Section 4.2.2, when enriching the document representation, we utilize WordNet to exploit hypernymy for clustering. We now demonstrate the effect of adding hypernyms into the datasets as follows.

Since FIHC obtained the best performance in terms of accuracy among the three comparing algorithms, we tested F²IDC and FIHC by the baseline method and the

addition of hypernyms of different levels. Table 4-6 shows the comparison of clustering results obtained by F²IDC and FIHC, respectively. In Table 4-6, “Baseline” means that no hypernyms are added; “H1” corresponds to the addition of direct hypernyms; “H2” stands for the addition of hypernyms of first and second levels, and so on. We chose the minimum support, ranging from 4% to 8% to run the baseline result of F²IDC for all datasets. The results in Table 4-6 show that FIHC decreases the clustering accuracy when increasing the levels of hypernyms. WordNet-based FIHC does not provide the improvement with respect to the baseline method. For the obtained results, the reasons could be:

- (1) Using hypernyms as additional features in the document enrichment process inevitably introduces a lot of noise into these datasets;
- (2) Word sense disambiguation was not performed to determine the proper meaning of each polysemous term in documents [24].

By Table 4-6, it is obvious that the average overall F-measure values of WordNet-based F²IDC are superior to that of WordNet-based FIHC when adding hypernyms of the first, second, and third levels on almost all datasets, except for WebKB dataset. The performance of F²IDC with the addition of direct hypernyms is better than that of F²IDC with higher levels of hypernyms on WebKB dataset. Due to the longer average length of documents in WebKB dataset, higher levels of hypernyms may add more noise to the clustering process and decrease the clustering accuracy.

In contrast to WordNet-based FIHC, our approach can ameliorate the effect of adding hypernyms by filtering out noise for clustering. The use of WordNet for F²IDC induces better clustering results on Classic dataset, while the improvements of the others are not particularly spectacular. In the case of the Reuters tasks, the limited improvement may not cause a particular worry. It is not likely to work well for text,

such as documents in Reuters-21578, which is guaranteed to be written in concise and efficiently [48].

Table 4-6: The effect of enriching the document representation.

| Datasets | <i>Classic</i> | | <i>Re0</i> | | <i>R8</i> | | <i>WebKB</i> | |
|-----------------|--------------------|-------------|--------------------|-------------|--------------------|-------------|--------------------|------|
| | F ² IDC | FIHC | F ² IDC | FIHC | F ² IDC | FIHC | F ² IDC | FIHC |
| Baseline | 0.54 | 0.51 | 0.50 | 0.40 | 0.55 | 0.55 | 0.44 | N.A. |
| H1 | 0.67 | 0.51 | 0.52 | 0.39 | 0.43 | 0.44 | 0.49 | N.A. |
| H2 | 0.65 | 0.50 | 0.51 | 0.38 | 0.43 | 0.44 | 0.48 | N.A. |
| H3 | 0.69 | 0.49 | 0.53 | 0.38 | 0.47 | 0.40 | 0.46 | N.A. |
| H4 | 0.66 | 0.47 | 0.53 | 0.38 | 0.47 | 0.40 | 0.45 | N.A. |
| H5 | 0.67 | 0.47 | 0.52 | 0.38 | 0.47 | 0.40 | 0.43 | N.A. |

To understand the reason why WordNet enhanced F²IDC to perform better, a sample of the cluster labels generated by F²IDC on Re0 dataset can be found in Table 4-7. Due to the rich semantic network representation provided by WordNet, F²IDC with WordNet generates more general and meaningful labels for clusters. For example, the label ‘commerce’ produced by F²IDC with WordNet is a more general concept than the labels ‘sell’ and ‘trade’ generated by F²IDC without WordNet.

Table 4-7: Cluster Labels generated by F²IDC algorithm on Re0 dataset.

| F ² IDC without WordNet | F ² IDC with WordNet |
|---|---|
| bank, dollar, currency, growth, industry, market, nation, rate, rise, rose, sell, trade | Activity, agent, assemblage, commerce, (commodity, good), currency, forecast, growth, merchant, nation, rate, record, (bush, rose, shrub) |

4.5.3.3. Sensitivity to Various Parameters

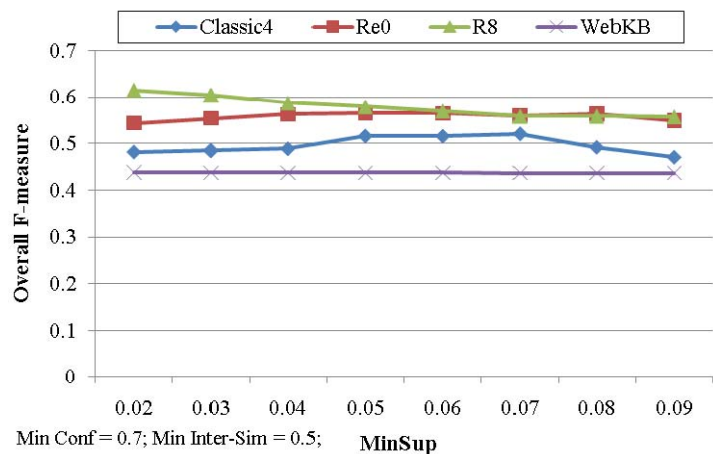
Figure 4-8(a) and (b) respectively depict the overall *F*-measure values of F²IDC and WordNet-based F²IDC when accepting different mandatory parameters, but

ignoring the parameter values of the optional ones. We observed that high clustering accuracies are fairly consistent while MinSup are set between 2% and 9% for F²IDC and set between 15% and 35% for WordNet-based F²IDC. As KClusters is not specified in each test case, the clusters merging step in Algorithm 3 has to decide the most appropriate number of output clusters, which are shown in Figure 4-8(b) and (d) for F²IDC and WordNet-based F²IDC, respectively.

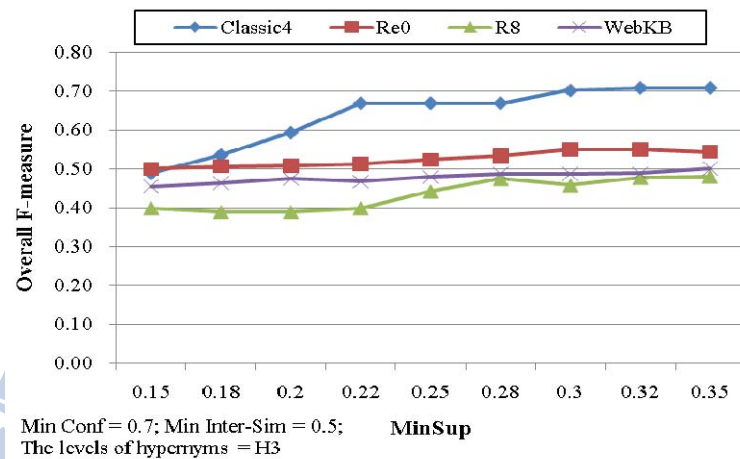
Based on our test, we concluded a general observation that the best choice of MinSup can be set between 4% and 8% for F²IDC, and set between 25% and 35% for WordNet-based F²IDC. Nevertheless, it cannot be over emphasized that MinSup should not be regarded as the only parameter for finding the optimal accuracy.

4.5.3.4. Efficiency and Scalability

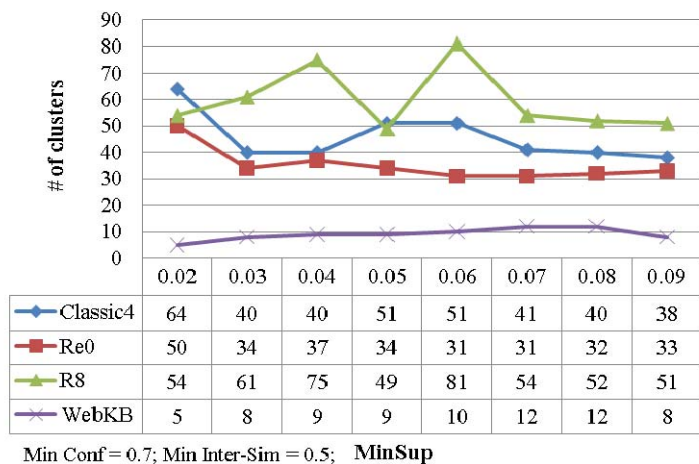
To analyze the scalability of our algorithm, we get 100,000 documents from RVC1 (Reuters Corpus Volume 1) dataset [31], which contains news from Reuters Ltd. There are three category sets: Topics, Industries, and Regions. In our experiments, we consider the Topics category set, which includes 23,149 training and 781,265 testing documents. Before clustering this dataset, documents were parsed by converting all terms in documents into lower case, removing stop words, and applying the stemming algorithm.



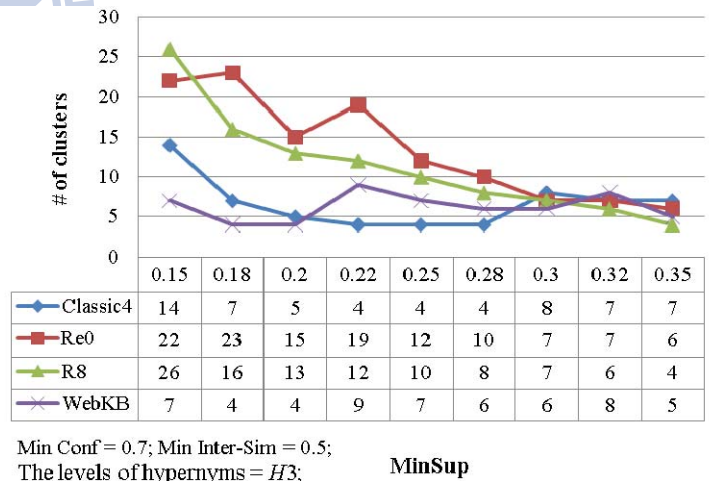
(a)



(b)



(c)



(d)

Figure 4-8: The accuracy test of F2IDC for different MinSup values with the optimal cluster numbers determined by the clusters merging step algorithm.

Figure 4-9 shows the runtimes with respect to the different sizes of RVC1 dataset, ranging from 10K to 100K documents, for different stages of our algorithm. The figure also shows that fuzzy association mining and initial clustering stages are the most two time-consuming stages in our algorithm. In the clustering process, most of the time is spent on constructing initial clusters and its runtime is almost linear with respect to the number of documents. As the efficiency of the fuzzy association rule mining is very sensitive to the input parameter MinSup, the runtime of F²IDC is inversely related to MinSup. In other words, runtime increases as MinSup decreases.

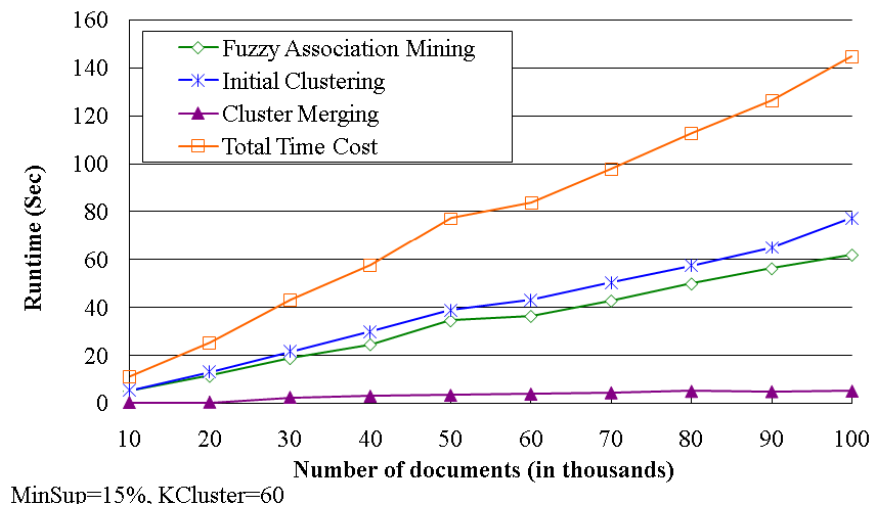


Figure 4-9: Scalability of F²IDC.

4.6 Summary

The importance of document clustering emerges from the massive volumes of textual documents created. Although numerous document clustering methods have been extensively studied in these years, there still exist several challenges for improving the clustering quality. Particularly, most of the current documents clustering algorithms, including FIHC, do not consider the semantic relationships among the terms. In this paper, we derived an effective Fuzzy Frequent Itemset-based

Document clustering (F^2 IDC) approach that combines fuzzy association rule mining with the external knowledge, WordNet, for grouping documents. The key advantage conferred by our proposed algorithm is that the generated clusters, labeled with conceptual terms, are easier to understand than clusters annotated by isolated terms. In addition, the extracted cluster labels may help for identifying the content of individual clusters.

Our experiments reveal that the proposed algorithm has better accuracy quality than that of FIHC, Bisecting k -means, and UPGMA methods on our datasets. Our primary findings are as follows:

- (1) Our approach facilitates the integration of the rich knowledge of WordNet into textual documents by effectively filtering out noise when adding hypernyms into documents and generating more conceptual labels for clusters.
- (2) FIHC performs better for documents of short average length, but worse for documents of long average length.
- (3) The other document clustering algorithms, like Bisecting k -means and UPGMA, are sensitive to the number of clusters.

In the next chapter, we will extend F^2 IDC to generate overlapping clusters for providing multiple subjective perspectives onto the same document to enhance its practical applicability.

Chapter 5

Fuzzy Frequent Itemset-based Soft Clustering (F²ISC) Approach

In this chapter, we further propose an effective Fuzzy Frequent Itemset-based Soft Clustering (F²ISC) approach by extending F²IDC under the consideration of overlapping cluster problem. F²ISC provides an accurate measure of confidence, and adopts the α -cut concept (defined in Definition 2.5) to assign each document to one or more than one target cluster.

Figure 5-1 shows the proposed F²ISC (Fuzzy Frequent Itemset-based Soft Clustering) framework, which consists of four modules, namely *Document Analysis Module*, *TermOnto Construction Module*, *Candidate Clusters Extraction Module*, and *Overlapping Clusters Generation Module* as explained in Sections 5.2.1, 5.2.2, 5.2.3, and 5.2.4, respectively.

In this framework, when receiving a set of textual documents, our first module will extract and select the key term set, and then the second module organizes it into a term forest (defined in Definition 4.2) by referring to WordNet for generating the Document Set D . The third module implements our fuzzy association rule mining procedure to generate the candidate cluster set. Finally, the last module constructs and evaluates the Document-Cluster Matrix (DCM) to produce the target clusters. The whole process will be illustrated by a comprehensive example.

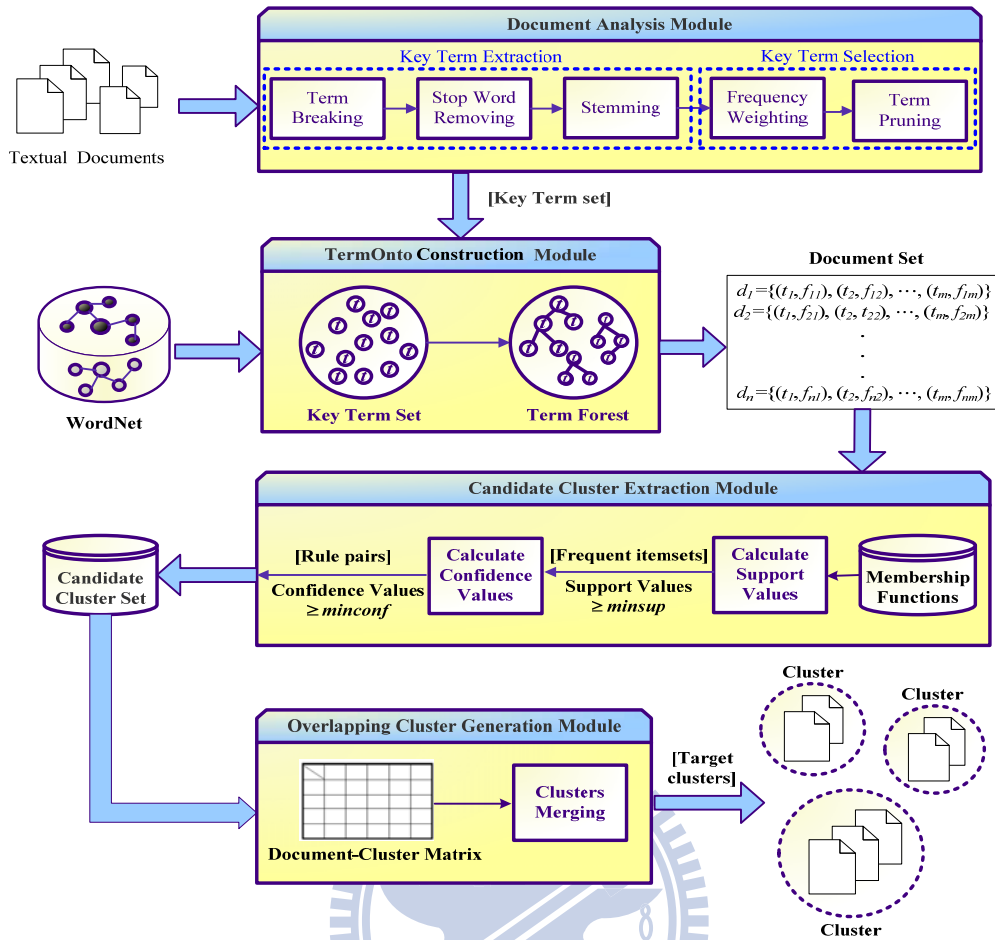


Figure 5-1: The F²ISC framework.

5.1 Document Analysis Module

There are two stages in this module, namely *Key Term Extraction* and *Key Term Selection*, for reducing the dimensionality of the source document set:

1. *Key Term Extraction*: The whole extraction process is as follows:
 - (1) First of all, each document is broken into sentences. Then, terms in each sentence are extracted as features. In this thesis, a term is regarded as the stem of a single word.

- (2) The terms appeared in a pre-defined stop-word list¹⁶ are removed.
 - (3) Remained terms are converted to their base forms by stemming. The terms with the same stem are combined for frequency counting. Finally, the frequency of each term in each document is recorded.
2. *Key Term Selection*: We understand that terms of low frequencies are supposed as noise and useless for identifying the appropriate cluster. Thus, we apply the tf-idf (term frequency \times inverse document frequency) method defined in Formula (4.1) to choose the key terms for the document set. A term will be discarded if its weight is less than a fixed tf-idf threshold γ . Subsequently, these retained terms form a set of key terms for the document set D , and we have defined them in Definitions 3.1 - 3.4.

5.2 TermOnto Construction Module

The objective of this module is based on the usage of WordNet for generating a richer document representation of the given document set. As the relationships of relevant terms have been predefined in WordNet ontology, in this module, we intend to use the hypernyms provided by WordNet ontology as useful features for document clustering. Thus, we use Algorithm 4.1, as shown in Figure 4-2, to generate the extended representation of each document for later mining process.

¹⁶ It contains a list of 571 stop words that was developed by the SMART project.

5.3 Candidate Cluster Extraction Module

After the above processes, documents are converted into structured term vectors. Then, the fuzzy data mining algorithm is executed to generate fuzzy frequent itemsets and output a candidate cluster set. In the module, we use the membership functions described in Figure 4-3 and the fuzzy association rule mining algorithm for texts shown in Figure 3-4 to generate the candidate cluster set.

5.4 Overlapping Cluster Generation Module

The objective of this module is to assign each document to multiple clusters $\{c_1^q, \dots, c_i^q\}$, where $i \geq 1$ and $q \geq 1$. The assignment process is based on the derived Document-Cluster matrix (DCM) defined in Definition 3.10. Then, we apply intersection of fuzzy set theory to compute the membership degree of each document in one candidate cluster with the other candidate clusters. Hence, we define one matrix, namely Multiple Clusters Matrix (MCM), in Definition 5.1.

Definition 5.1: A *Multiple Clusters Matrix (MCM)*, denoted $M = [m_{ig}]$, is an $n \times C_k^2$ matrix, such that $m_{ig} = \min\{m_{il}, m_{ij}\}$ is the membership degree of document d_i in intersection of two candidate clusters $\tilde{c}_l^q \cap \tilde{c}_j^q$, where $l, j \in \{1, 2, \dots, k\}$, $l \neq j$, and $q = 1$. A formal illustration of MDM can be found in Figure 5-2.

$$M = \begin{matrix} & \tilde{c}_1^1 \cap \tilde{c}_2^1 & \tilde{c}_1^1 \cap \tilde{c}_3^1 & \dots & \tilde{c}_{k-1}^1 \cap \tilde{c}_k^1 \\ \begin{matrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{matrix} & \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1C_k^2} \\ m_{21} & m_{22} & \dots & m_{2C_k^2} \\ \vdots & \ddots & \ddots & \vdots \\ m_{n1} & m_{n2} & \dots & m_{nC_k^2} \end{bmatrix} \end{matrix} \Big]_{n \times C_k^2}$$

Figure 5-2: A formal illustration of Multiple Clusters Matrix.

Moreover, we apply the α -cut threshold [64][68] determined by Formula (5.1) to evaluate the minimum value which satisfies the restrictive condition, and it can appropriately provide flexibility to overlapping clusters.

$$\alpha < \min_{1 \leq g \leq C_k^2} \left\{ \max_{1 \leq i \leq n} [m_{ig}] \right\} \quad (5.1)$$

Then, based on the obtained DCM, an unassigned document d_i might belong to more than one target cluster by using Formula (5.2).

$$c_l^q = \left\{ d_i \mid v_{il} > \max\{(\rho - \alpha), \alpha\} \text{ where } \rho = \max\{v_{i1}, v_{i2}, \dots, v_{ik}\} \in \tilde{c}_l^q \right\} \quad (5.2)$$

Finally, to avoid low clustering accuracy, the inter-cluster similarity(defined by Formula (3.9) in Chapter 3) between two target clusters is calculated to merge the small target clusters with the similar topic.

Algorithm 5.1 shown in Figure 5-3 is used to assign each document to the fitting target clusters, and finally builds a target cluster set for output.

5.5 An Illustrative Example of F²ISC Method

Suppose we have a document set $D = \{d_1, d_2, \dots, d_5\}$ and its key term set $K_D = \{\text{sale, trade, medical, health}\}$. Figure 5-4 illustrates the process of Algorithm 4.1 to

obtain the representation of all documents. Moreover, rectangle nodes represent actual key terms appearing in the document set; spheroid nodes represent newly-added hypernyms. In this example, the key term ‘sale’ has the parent nodes ‘marketing’ and ‘commerce’. Similarly, ‘trade’ and ‘marketing’ have the same parent node ‘commerce’.

Algorithm 5.1. Basic algorithm to obtain the target clusters

Input: A document set $D = \{d_1, d_2, \dots, d_i, \dots, d_n\}$; The key term set $K_D = \{t_1, t_2, \dots, t_j, \dots, t_m\}$; The candidate cluster set $\tilde{C}_D = \{\tilde{c}_1^1, \dots, \tilde{c}_{l-1}^1, \tilde{c}_l^1, \dots, \tilde{c}_k^q\}$; A minimum *Inter-Sim* threshold δ ;

Output: The target cluster set $C_D = \{c_1^1, c_2^1, \dots, c_i^q, \dots, c_f^q\}$

1. Build $n \times p$ document-term matrix $W = [w_{ij}^{\max-R_j}]$. // $w_{ij}^{\max-R_j}$ is the weight (fuzzy value) of t_j in d_i and $t_j \in L_1$.

2. Build $p \times k$ term-cluster matrix $G = [g_{jl}^{\max-R_j}]$. // $g_{jl}^{\max-R_j} = \frac{\text{score}(\tilde{c}_l^q)}{\sum_{i=1}^n w_{ij}^{\max-R_j}}$, $1 \leq j \leq p$, $1 \leq l \leq k$,

and, $\text{score}(\tilde{c}_l^q) = \sum_{d_i \in \tilde{c}_l^q, t_j \in \tau} w_{ij}^{\max-R_j}$, where $w_{ij}^{\max-R_j}$ is the weight (fuzzy value) of t_j in

d_i and $t_j \in L_1$.

3. Build $n \times k$ document-cluster matrix $V = W \cdot G = [v_{il}] = \sum_{p=1}^p w_{ip} g_{pl}$.

4. Build $n \times C_2^k$ multiple clusters matrix $M = [m_{ig}]$

5. Decide the α -cut threshold $\alpha < \min_{1 \leq g \leq C_2^k} \left\{ \max_{1 \leq i \leq n} [m_{ig}] \right\}$

4. Based on V , assign d_i to target clusters

$$c_i^q = \{d_i \mid v_{il} > \max\{(\rho - \alpha), \alpha\} \text{ where } \rho = \max\{v_{i1}, v_{i2}, \dots, v_{ik}\} \in \tilde{c}_l^q\}$$

6. Clusters merging

(1) For each $c_i^q \in C_D$ do

(a) If ($c_i^q = \text{null}$) then { remove this target clusters c_i^q from C_D }

(2) For each pair of target clusters $(c_x^q, c_y^q) \in C_D$ do

(a) Calculate the *Inter_sim*

(b) Store the results in the Inter-Cluster Similarity matrix I .

(3) If (one of the *Inter_sim* value in $I \geq \delta$) then

(a) Select (c_x^q, c_y^q) with the highest *Inter_sim*.

(b) Merge the smaller target cluster into the larger target cluster.

(c) Repeat Step (2) to update I

7. Output C_D

Figure 5-3: The detailed description of Algorithm 5.1.

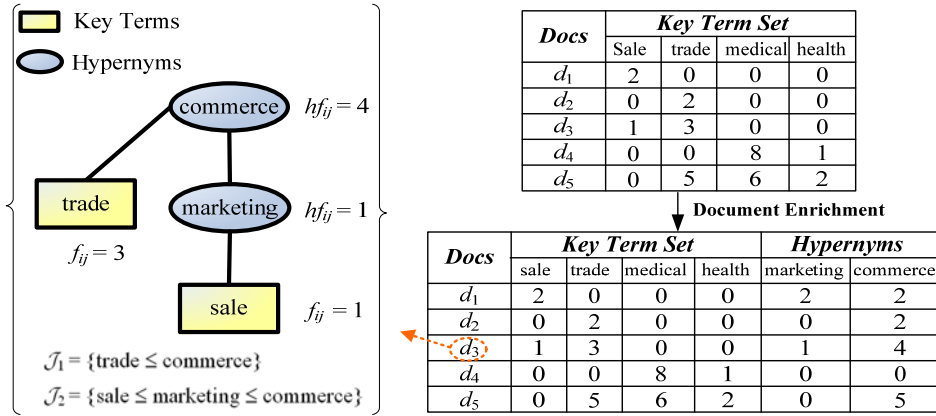


Figure 5-4: The process of Algorithm 4.1 of this example.

Consider the representation of all documents generated from Figure 5-4, the membership functions defined in Figure 4-3, the minimum support value 80%, and the minimum confidence value 80% as inputs. The fuzzy frequent itemsets discovery procedure is depicted in Figure 5-5.

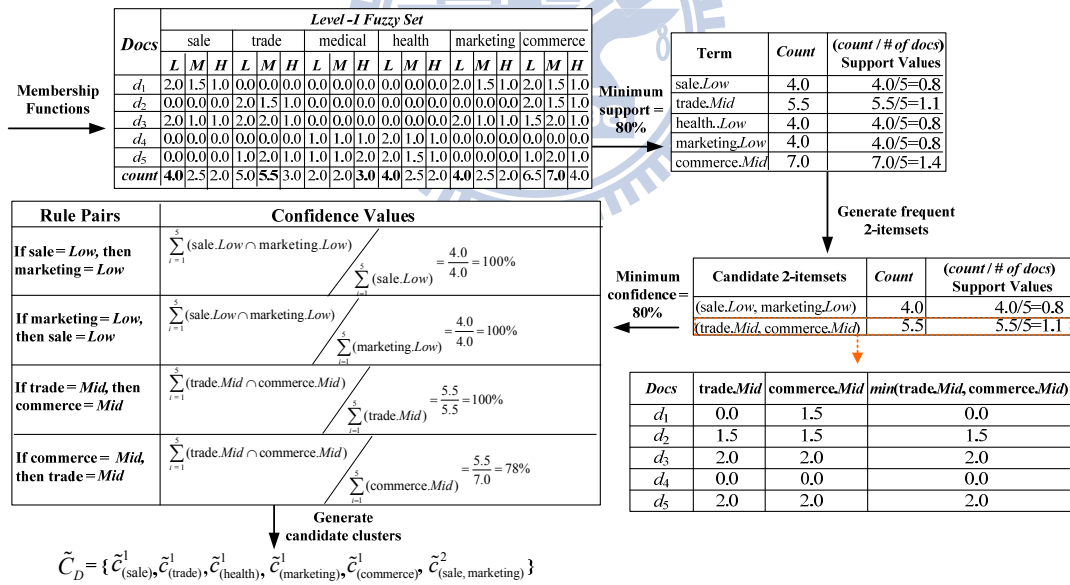


Figure 5-5: The process of Algorithm 4.2 of this example.

Moreover, consider the candidate cluster set \tilde{C}_D was already generated in Figure 5-5. Now, suppose the minimum *Inter-Sim* value is 0.5. Figure 5-6 illustrates the process of Algorithm 5.1, together with the final results.

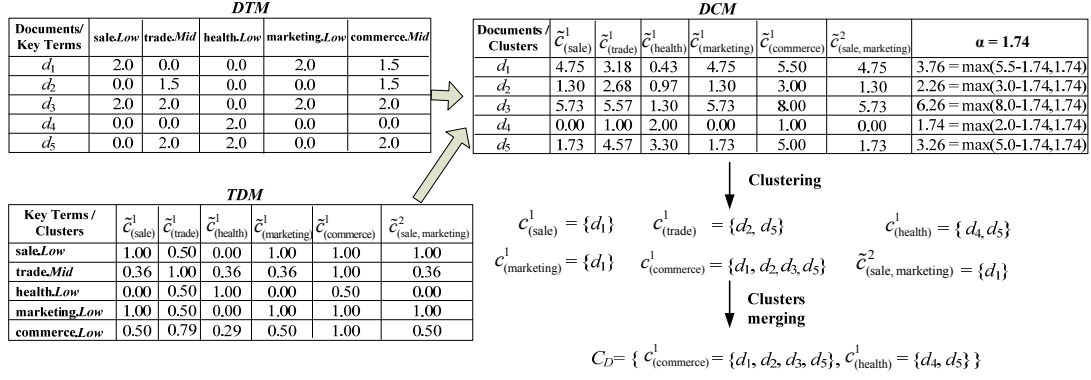


Figure 5-6: The process of Algorithm 5-1 of this example.

5.6 Experiments

In this section, we experimentally evaluated the performance of the proposed algorithm by comparing with that of FIHC, k -means, Bisecting k -means, and UPGMA algorithms. To test the proposed approach, we used four different kinds of datasets: *Classic*, *Re0*, *R8*, and *WebKB*, which are described in Subsection 4.3.1 and summarized the statistics in Table 4-1.

Notice that overall F-Measure favors for the hard assignment generated by clustering algorithms. In order to demonstrate the performance of our approach, we present experiments in which we generated hard assignment (this has been called *hardening* the clusters) [2] and then evaluated the output of our algorithm. The hardening scheme is simply performed by assigning each document to the cluster which has a maximum membership degree among all the document clusters. Thus, it can be employed to evaluate the performance of our approach by comparing with the other hard clustering methods. Thus, we use overall F-Measure to evaluate the clustering quality of F^2 ISC and the other compared algorithms.

5.6.1 Parameters Selection

Table 5-1 summarizes the parameters for our proposed method and the other algorithms to compare the clustering performance. Since k -means, Bisecting k -means, and UPGMA may generate different clustering results each time with randomly chosen initial value. Therefore, the final result of these three algorithms is an average from five runs performed on a given dataset.

Table 5-1: List of all parameters for our algorithms and the other four algorithms.

| Parameter Name | F ² ISC | FIHC | k -means | Bi. k -means | UPGMA |
|-----------------------------|--------------------------------|--------|------------|-------------------|--------|
| Datasets | Classic, Re0, R8, WebKB | | | | |
| Stopword Removal | Yes | | | | |
| Stemming | Yes | | | | |
| Length of the smallest term | Three | | | | |
| Weight of the term vector | TF | tf-idf | tf-idf | tf-idf | tf-idf |
| Levels of hypernyms | h_1, h_2, h_3, h_4, h_5 | | | | |
| Cluster count k | 5, 10, 15, 30, 45, 60, 80, 100 | | | | |

Before applying F²ISC, we first consider the feature selection strategy. In order to select the most representative features, we use Formula (4.1) to obtain the key terms with weights higher than the pre-defined thresholds γ . Table 4-3 shows the keyword statistics of our test datasets and the suggested thresholds for each dataset.

The two algorithms, F²ISC and FIHC, all have two main parameters for the adjustment of accuracy quality. This first one is mandatory and is denoted MinSup, which means the minimum support for frequent itemsets generation. The other one is optional, and is denoted KCluster, which represents the number of clusters.

5.6.2 Experimental Results and Analysis

The experiments were conducted by the following steps. First, we evaluated our approach, F²ISC, on the four selected datasets described in Section 4.1 and compared its accuracy with that of FIHC, the standard k -means, Bisecting k -means, and UPGMA. Second, we verified if the use of WordNet can improve the clustering accuracy on these compared algorithms and generated conceptual labels for the derived clusters. Third, the dataset Reuters was chosen to evaluate the efficiency and scalability of F²ISC.

5.6.2.1. Comparison of F²ISC with Other Algorithms

Figure 5-7 presents the obtained overall F-Measure values for F²ISC and the other algorithms by comparing eight different numbers of clusters on four datasets. For each algorithm, we run each dataset enriched with the top 5 levels of hypernyms. We tested each algorithm's clustering results with the value h , the levels of hypernyms, from 1 to 5 and selected the best results. We chose the MinSup threshold from the elements in {25%, 28%, 30%, 32%, 35%} to run F²ISC with WordNet for all datasets. Moreover, we use the minimum support, ranging from 3% to 6% for FIHC for all datasets. Notice that UPGMA is not available for large data sets because some experimental results cannot be generated for UPGMA. Since FIHC is not available for the documents of long average length, there is no experimental result generated on the WebKB dataset.

By Table 5-2, it is obvious that the average overall F-measure values of F²ISC with WordNet are superior to that of the other algorithms on all datasets. Although the

average accuracy of Bisecting k -means and FIHC shown in Figure 5-7 are slightly better than that of F²ISC in several cases. We argue that the exact number of clusters in a document set is usually unknown in real case, and F²ISC is robust enough to produce stable, consistent and high quality clusters for a wide range number of clusters. This can be realized by observing the average overall F -measure values of all test cases. From Figure 5-7, we also observed that the clustering accuracy of k -means, Bisecting k -means, and UPGMA are sensitive when the number of clusters changes. These algorithms require users to specify the number of cluster as an input parameter, which may imply poor clustering accuracy when we input an incorrect parameter [17].

Table 5-2: Average overall F-measure comparison for five clustering algorithms on the four datasets.

| Datasets | F ² ISC(h) | FIHC(h) | k -means(h) | Bi. k -means (h) | UPGMA(h) |
|----------|---------------------------|-------------|-------------------|------------------------|--------------|
| Classic | 0.65(3) * | 0.49(1) | 0.47(2) | 0.45(5) | N.A. |
| Re0 | 0.53(3) * | 0.36(1) | 0.35(2) | 0.34(5) | 0.36(1) |
| R8 | 0.44(3) * | 0.42(1) | 0.34(3) | 0.33(3) | N.A. |
| WebKB | 0.48(1) * | N.A. | 0.16(4) | 0.15(1) | 0.38(1) |

N.A. means not scalable to run * means the best competitor

5.6.2.2. The effect of the Enriched Document Representation

As described in the second module of our approach, when enriching the document representation, we use the hypernyms from WordNet as useful features for clustering. We demonstrate the effect of adding hypernyms in our approach. In the following, all algorithms are tested by the baseline method and the addition of hypernyms of various levels.

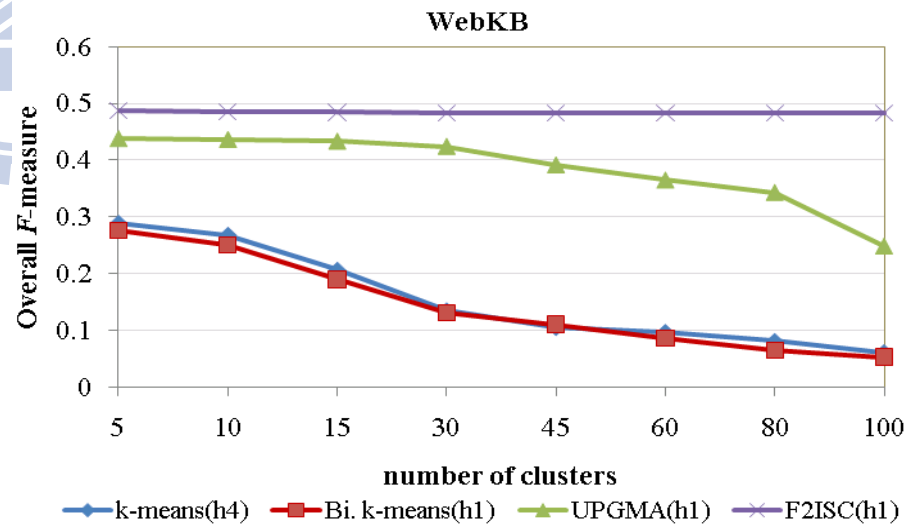
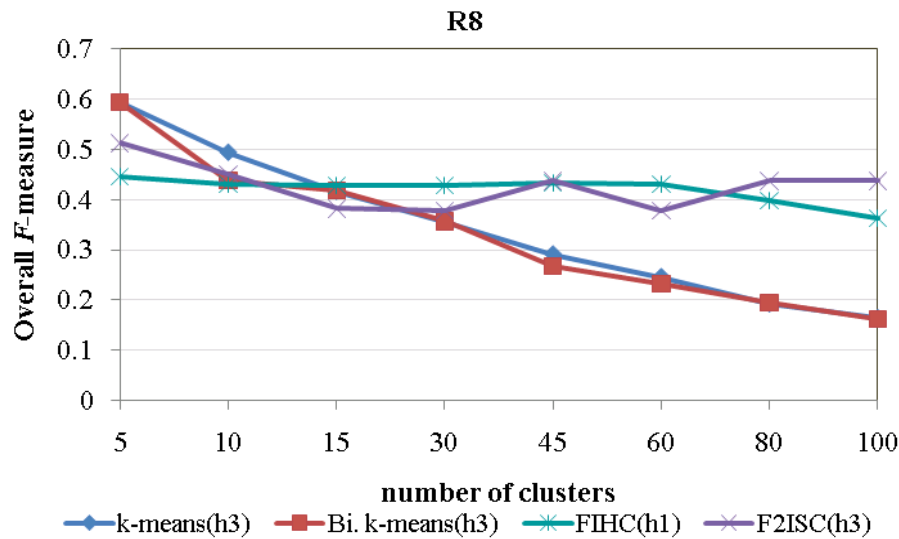
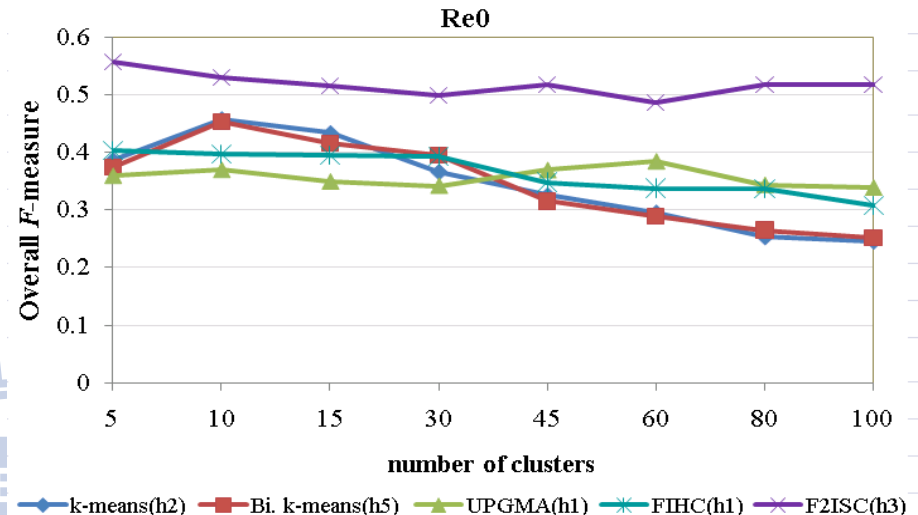
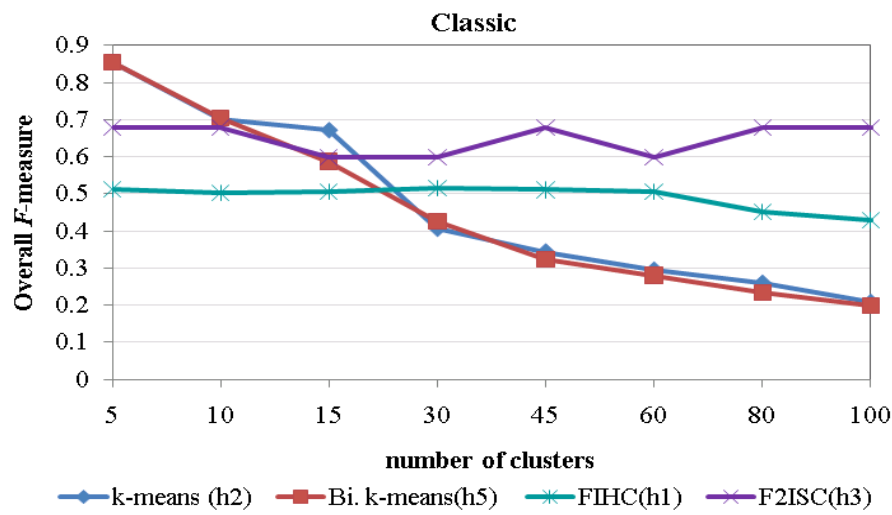


Figure 5-7: Overall F-measure comparison for five clustering algorithms on the four datasets.

Table 5-3 shows the average overall F-measure results obtained by all algorithms on classic and re0 datasets. The results for R8 and WebKB datasets are shown in Table 5-4. In Table 5-3 and Table 5-4, “Baseline” means that no hypernyms are added; “ h_1 ” corresponds to the addition of direct hypernyms; “ h_2 ” stands for the addition of hypernyms of first and second levels, and so on. We chose the minimum support values, ranging from 4% to 8%, to run the baseline result of F²ISC for all datasets. The evaluation results in Table 5-3 and Table 5-4 confirm that the average overall F-measure values of WordNet-based F²ISC performance are superior to that of the other algorithms when adding hypernyms of the first, second, and third levels on almost all datasets, except for WebKB dataset. The performance of F²ISC with the addition of direct hypernyms is better than that of F²ISC with higher levels of hypernyms on WebKB dataset. Due to the longer average length of documents in WebKB dataset, we think that higher levels of hypernyms may add more noise to the clustering process and decrease the clustering accuracy.

From Table 5-3 and Table 5-4, the use of WordNet for F²ISC induces better clustering results at least 5% higher than the other algorithms on Classic and WebKB datasets, particularly the improvement of Classic dataset. However, adding hypernyms may not be beneficial for the clustering task. The reason is that using hypernyms as additional features in the document enrichment process inevitably introduces a lot of noise into these datasets. In contrast to the other WordNet-based algorithms, our approach can ameliorate the effect of adding hypernyms by filtering out noise for clustering on Classic and WebKB datasets.

Table 5-3: The effect of enriching the document representation on *Classic* and *Re0* datasets.

| Datasets | <i>Classic</i> | | | | | <i>Re0</i> | | | | |
|-----------------------|--------------------|-------------|-------------|-------------|-------|--------------------|-------------|-------------|-------------|-------------|
| | F ² ISC | FIHC | k-means | Bi. k-means | UPGMA | F ² ISC | FIHC | k-means | Bi. k-means | UPGMA |
| Baseline | 0.48 | 0.47 | 0.45 | 0.46 | N.A. | 0.55 | 0.38 | 0.36 | 0.35 | 0.40 |
| <i>h</i> ₁ | 0.63 | 0.49 | 0.46 | 0.44 | N.A. | 0.52 | 0.36 | 0.34 | 0.34 | 0.36 |
| <i>h</i> ₂ | 0.64 | 0.49 | 0.47 | 0.44 | N.A. | 0.52 | 0.35 | 0.35 | 0.34 | 0.35 |
| <i>h</i> ₃ | 0.65 | 0.48 | 0.47 | 0.45 | N.A. | 0.53 | 0.36 | 0.35 | 0.34 | 0.35 |
| <i>h</i> ₄ | 0.61 | 0.45 | 0.45 | 0.44 | N.A. | 0.51 | 0.36 | 0.35 | 0.34 | 0.35 |
| <i>h</i> ₅ | 0.62 | 0.45 | 0.45 | 0.45 | N.A. | 0.51 | 0.36 | 0.33 | 0.34 | 0.35 |

N.A. means not scalable to run boldface entries highlight the best competitor in each column from *h*₁ to *h*₅ (the row headings)

Table 5-4: The effect of enriching the document representation on *R8* and *Webkb* datasets.

| Datasets | <i>R8</i> | | | | | <i>Webkb</i> | | | | |
|-----------------------|--------------------|-------------|-------------|-------------|-------|--------------------|------|-------------|-------------|-------------|
| | F ² ISC | FIHC | k-means | Bi. k-means | UPGMA | F ² ISC | FIHC | k-means | Bi. k-means | UPGMA |
| Baseline | 0.53 | 0.52 | 0.35 | 0.34 | N.A. | 0.43 | N.A. | 0.15 | 0.15 | 0.35 |
| <i>h</i> ₁ | 0.36 | 0.42 | 0.34 | 0.33 | N.A. | 0.48 | N.A. | 0.15 | 0.15 | 0.38 |
| <i>h</i> ₂ | 0.37 | 0.41 | 0.34 | 0.33 | N.A. | 0.43 | N.A. | 0.15 | 0.14 | 0.38 |
| <i>h</i> ₃ | 0.44 | 0.37 | 0.34 | 0.33 | N.A. | 0.37 | N.A. | 0.15 | 0.14 | 0.38 |
| <i>h</i> ₄ | 0.43 | 0.37 | 0.33 | 0.33 | N.A. | 0.33 | N.A. | 0.16 | 0.14 | 0.38 |
| <i>h</i> ₅ | 0.43 | 0.36 | 0.33 | 0.32 | N.A. | 0.33 | N.A. | 0.15 | 0.14 | 0.38 |

N.A. means not scalable to run boldface entries highlight the best competitor in each column from *h*₁ to *h*₅ (the row headings)

However, comparing with the baseline method, the use of WordNet decreases the clustering accuracy on Re0 and R8 datasets for our approach and the other compared algorithms. For the obtained results, the reasons could be:

- (1) It is not likely to work well for text, such as documents in Reuters-21578, which is guaranteed to be written in concise and efficiently [48].
- (2) Word sense disambiguation was not performed to determine the proper meaning of each polysemous term in documents [24].

5.6.2.3. Efficiency and Scalability

Our algorithm, F²ISC, involves three major phases: finding fuzzy frequent itemsets, initial clustering, and clusters merging. Figure 5-8 shows the scalabilities of

F²ISC on different sizes of Reuters datasets, ranging from 1K to 8K documents.

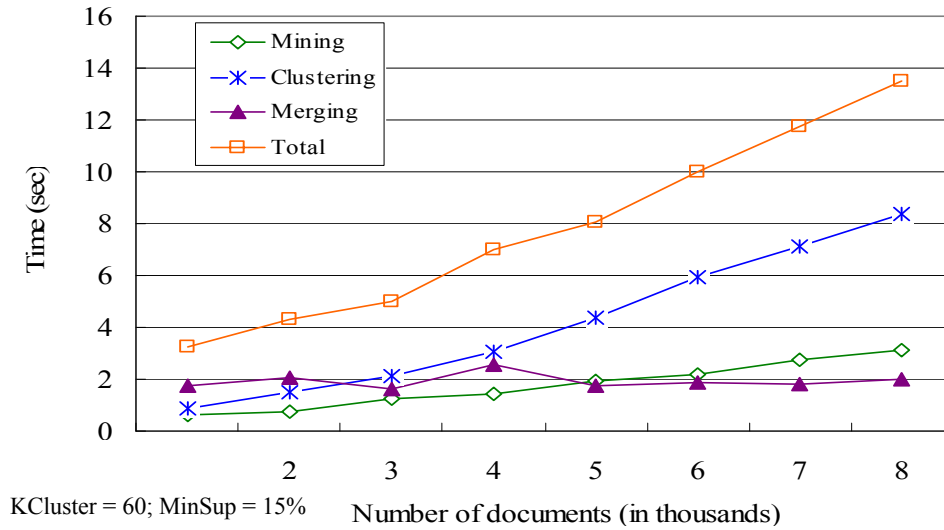


Figure 5-8: The detailed time cost analysis of F²ISC on Reuters dataset.

5.7 Summary

In this chapter, we derived a fuzzy-based document clustering approach that combines fuzzy association rule mining with WordNet to take semantic information into account. In the total processes, we begin with the process of document pre-processing and further enrich the initial representation of all documents by using hypernyms of WordNet in order to exploit the semantic relations between terms. Then, fuzzy association data mining algorithm automatically generates fuzzy frequent itemsets and regards them as candidate clusters. Finally, each document is dispatched into more than one cluster by referring to these candidate clusters, and then highly similar clusters are merged.

Moreover, document clustering methods should provide multiple subjective perspectives onto the same document to enhance their practical applicability. For this issue, we adopt the α -cut concept in the process of document clustering to assign each

document to one or more than one target cluster. The generated overlapping clusters occur naturally in many applications such as Yahoo! directory

Our experiments reveal that the proposed algorithm has better cluster quality than that of FIHC, k -means, Bisecting k -means, and UPGMA methods based on the four datasets of Classic, Re0, R8, and WebKB.



Chapter 6

Conclusions and Future Work

6.1 Conclusions

The importance of document clustering emerges from the massive volumes of textual documents created. Although numerous document clustering methods have been extensively studied in these years, there still exist several challenges for increasing the clustering quality. Particularly, most of the current document clustering algorithms do not consider the semantic relationships among the terms nor search an organization of documents into overlapping clusters. In this thesis, we derived three fuzzy frequent itemset-based document clustering methods, namely F^2IHC , F^2IDC , and F^2ISC , to solve these challenges.

The key advantage conferred by our proposed algorithms, F^2IDC and F^2ISC , is that the generated clusters, labeled with conceptual terms, are easier to understand than clusters annotated by isolated terms. In addition, the extracted cluster labels may help for identifying the content of individual clusters. Moreover, the other advantage of F^2ISC method is that overlapping clusters occur naturally in many applications such as Yahoo! directory.

Our experiments reveal that the proposed algorithm has better accuracy quality than that of $FIHC$, k -means, Bisecting k -means, and UPGMA methods based on the comparison on these datasets. Our primary findings are as follows:

- (1) The use of fuzzy association rule mining discovery important candidate clusters for document clustering to increase the accuracy quality of document clustering.

- (2) F²IDC and F²ISC approach are successful in avoiding the expansion of terms with noisy features on Classic and WebKB datasets.
- (3) FIHC performs better for documents of short average length, but worse for documents of long average length.
- (4) The other document clustering algorithms, like *k*-means, Bisecting *k*-means, and UPGMA, are sensitive when the number of clusters changes.

6.2 Future Work

Our future work will focus on the following two aspects:

- (1) Combining the syntactic analysis: For finding the important terms in a document, terms with different part-of-speech (POS) and syntactic attributes should be set different weights according to their relatedness in a document [67]. There are a lot of syntactic analysis tools that can be used to tag all terms in the document set, i.e., Qtag¹⁷ parser. We will further study whether our proposed algorithm with a syntactic analysis tool can improve the clustering results.
- (2) Incrementally updating the cluster tree: When the number of documents increases sequentially in a document set, it is inefficient to reform the cluster tree for each new insertion. That is, it is admirable to reflect the current state of the whole document set by incrementally updating the cluster tree [14][43]. Therefore, we intend to propose an efficient incremental clustering algorithm for assigning a new document to the most similar existing cluster in the future. Some recent researches on data mining concerning data streaming [41][18][25] may be applicable for such incremental clustering development.

¹⁷ <http://www.english.bham.ac.uk/staff/omason/software/qtag.html>

(3) Using Wikipedia: we will consider the abundant structural relation within Wikipedia, such as hyperlinks and hierarchical categories, to improve the performance of clustering [57]. In addition, we will further compare our proposed approaches with other new frequent itemset-based document algorithms, such as Clustering based on Frequent Word Sequences (CFWS) [32] and Maximum Capturing (MC) [66].



Bibliography

- [1] Agrawal, R., Imielinski, T., A. Swami, Mining association rules between sets of items in large databases, *In: Proc. of ACM SIGMOD Int'l Conf. on Management of Data*, 1993, pp.207-216.
- [2] Andrews, N. O., Fox, E. A., *Recent Developments in Document Clustering*, Technical Report TR-07-35, Computer Science, Virginia Tech, 2007.
- [3] Beil, F., Ester, M., Xu, X., Frequent term-based text clustering, *In: Proc. of Int'l Conf. on knowledge Discovery and Data Mining (KDD'02)*, 2002, pp. 436-442.
- [4] Bellot, P., El-Beze, M., *A Clustering Method for Information Retrieval*, Technical Report IR-0199, 1999.
- [5] Chen, C. L., Tseng, F. S. C., and Liang, T., An integration of fuzzy association rules and WordNet for document clustering, *Knowledge and Information Systems (KAIS)*, Revision Submitted, 2010/03/20.
- [6] Chen, C. L., Tseng, F. S. C., and Liang, T., An integration of fuzzy association rules and WordNet for document clustering, *In: Proc. of the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-09)*, 2009, pp. 147-159.
- [7] Chen, C. L., Tseng, F. S. C., and Liang, T., An integration of WordNet and fuzzy association rule mining for multi-label document clustering, *Data and Knowledge Engineering*, to appear.
- [8] Chen, C. L., Tseng, F. S. C., and Liang, T., Hierarchical document clustering based on fuzzy association rule mining, *In: Proc. of the 3rd International Conference on Innovative Computing Information and Control, (ICICIC'08)*, 2008/06, pp. 326-330.

- [9] Chen, C. L., Tseng, F. S. C., and Liang, T., Mining fuzzy frequent itemsets for hierarchical document clustering, *Information Processing and Management*, Vol. 46, No. 2, March 2010, pp. 193-211.
- [10] Craven, M., DiPasquo, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S., Learning to extract symbolic knowledge from the world wide web, *In: AAAI-98*, 1998.
- [11] Dave, K., Lawrence, S., Pennock, D. M., Mining the peanut gallery: opinion extraction and semantic classification of product reviews, *In: Proc. of the 12th Int'l Conf. on World Wide Web*, 2003.
- [12] de Campos, L. M., Moral, S., Learning rules for a fuzzy inference model, *Fuzzy Sets and Systems*, Vol. 59, 1993, pp. 247-257.
- [13] Delgado, M., Martan-Bautista, M. J., Sanchez, D., Vila, M. A., Mining text data: special features and patterns, *In: Proc. of EPS Exploratory Workshop on Pattern Detection and Discovery in Data Mining*, 2002, pp. 140-153.
- [14] Exarchos, T. P., Tsipouras, M. G., Papaloukas, C., Fotiadis, D. I., An optimized sequential pattern matching methodology for sequence classification, *Knowledge and Information Systems*, Vol. 19, No. 2, 2009, pp. 249-264.
- [15] Feldman, R., Dagan, I., Knowledge discovery in textual databases (KDT), *In: Proc. of the 1st ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 1995, pp. 112-117.
- [16] Fung, B. C. M., *Hierarchical Document Clustering Using Frequent Itemset*, Master thesis, Simon Fraser University, 2002.
- [17] Fung, B., Wang, K., Ester, M., Hierarchical document clustering using frequent itemsets, *In: Proc. of SIAM Int'l Conf. on Data Mining (SDM'03)*, May 2003, pp. 59-70.

- [18] Guha, S., Meyerson, A., Mishra, N., Motwani, R., O’Callaghan, L., Clustering data streams: theory and practice, *IEEE Trans. on Knowledge and Data Eng.*, Vol. 15, No. 3, 2003, pp. 515–528.
- [19] Han, E. H., Boley, B., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., Moore, J., Webace: A web agent for document categorization and exploration, *In: Proc. of the 2nd Int’l Conf. on Autonomous Agents*, 1998, pp. 408-415.
- [20] Hipp, J., Guntzer, U., Nakhaeizadeh, G., Algorithms for association rule mining - a general survey and comparison, *ACM SIGKDD Explorations Newsletter*, Vol. 2, 2000, pp. 58–64.
- [21] Hong, T. P., Lee, Y. C., An overview of mining fuzzy association rules, *In: H. Bustince et al., (eds.), Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*, 2008, pp. 397-410.
- [22] Hong, T. P., Lin, K. Y., Wang, S. L., Fuzzy data mining for interesting generalized association rules, *Fuzzy Sets and Systems*, Vol. 138, No. 2, 2003, pp. 255-269.
- [23] Hotho, A., Maedche, A., Staab, S., Ontology-based textual document clustering, *Kunstliche Intelligenz*, Vol. 16, No. 4, 2002, pp. 48–54.
- [24] Hotho, A., Staab, S., Stumme, G., Wordnet improves textual document clustering, *In: Proc. of SIGIR Int’l Conf. on Semantic Web Workshop*, 2003.
- [25] Huang, Z., Sun, S., Wang, W., Efficient mining of skyline objects in subspaces over data streams, *Knowledge and Information Systems*, Vol. 22, No. 2, 2010. pp. 159-183.
- [26] Jain, A. K., Dubes, R. C., *Algorithms for clustering data*, Prentice-Hall, Inc., 1988.

- [27] Jing, L., *Survey of text Clustering*, [http://www.alphaminer.org/document/downloads/textmining/survey of text clustering.pdf](http://www.alphaminer.org/document/downloads/textmining/survey%20of%20text%20clustering.pdf), 2008.
- [28] Jing, L., Zhou, L., Ng, M. K., Huang, J. Z., Ontology-based distance measure for text clustering, *In: Proc. of SIAM Int'l Conf. on Data Mining*, 2006.
- [29] Kaufman, L., Rousseeuw, P. J., *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley and Sons, Inc., 1990.
- [30] Kaya, M., Alhajj, R., Utilizing genetic algorithms to optimize membership functions for fuzzy weighted association rule mining, *Applied Intelligence*, Vol. 24, No. 1, 2006, pp. 7-15.
- [31] Lewis, D. D., Yang, Y., Rose, T. G., Li, F., RCV1: a new benchmark collection for text categorization research, *Journal of Machine Learning Research*, Vol. 5, 2004, pp. 361 - 397.
- [32] Li, Y. J., Chung, S. M., Holt, J. D., Text document clustering based on frequent word meaning sequences, *Data and Knowledge Engineering*, Vol. 64, 2008, pp. 381-404.
- [33] Lin, K., Kondadadi, R., A word-based soft clustering algorithm for documents, *Computers and Their Applications*, 2001, pp. 391-394.
- [34] Lin, S. H., Shih, C.S., Chen, M. C., Ho, J. M., Ko, M. T., Huang, Y. M., Extracting classification knowledge of internet documents with mining term association: a semantic approach, *In: Proc. of the 21st Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1998, pp. 241-249.
- [35] Liu, B., Hsu, W., Ma, Y., Pruning and summarizing the discovered associations, *In: Proc. of the ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, 1999, pp. 125-134.

- [36] MacQueen, J. B., Some methods for classification and analysis of multivariate observations, *In: Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281-297.
- [37] Mandhani, B., Joshi, S., Kummamuru, K., A matrix density based algorithm to hierarchically co-cluster documents and words, *In: Proc. of the 12th Int'l Conf. on World Wide Web*, 2003, pp. 511-518.
- [38] Martín-Bautista, M. J., Sánchez, D., Chamorro-Martínez, J., Serrano, J. M., Vila, M. A., Mining web documents to find additional query terms using fuzzy association rules, *Fuzzy Sets and Systems*, Vol. 148, No. 1, 2004, pp.85-104.
- [39] Michenerand, C. D., Sokal, R. R., A quantitative approach to a problem in classification, *Evolution*, Vol. 11, 1957, pp. 130-162.
- [40] Miller, G.A., WordNet: a lexical database for English, *J. Communications of the ACM*, Vol. 38, No. 11, 1995, pp. 39-41.
- [41] Ordóñez, C., Clustering binary data streams with k -means, *In: Proc. of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2003, pp. 2-19.
- [42] Özgür, A., Güngör, T., Classification of skewed and homogeneous document corpora with class-based and corpus-based keywords, *In: Proc. of the 29th German Conf. on Artificial Intelligence*, 2006, pp.91-101.
- [43] Pons-Porrata, A., Berlanga-Llavori, R., Ruiz-Shulcloper, J., Topic discovery based on text mining techniques, *Information Processing and Management*, Vol. 43, No. 3, 2007, pp. 752-768.
- [44] Porter, M. F., An algorithm for suffix stripping, *Program*, Vol. 14, No. 3, 1980, pp. 130-137.

- [45] Recupero, D. R., A new unsupervised method for document clustering by using WordNet lexical and conceptual relations, *Information Retrieval*, Vol. 10, No. 6, 2007, pp. 563-579.
- [46] Salton, G., Buckley, C., Term-weighting approaches in automatic text retrieval, *Information Processing and Management*, Vol. 24, No. 5, 1988, pp. 513-523.
- [47] Salton, G., Wong, A., Yang, C., A vector space model for automatic indexing, *Communications of the ACM*, Vol. 18, No. 11, 1975, pp. 613-620.
- [48] Scott, S., Matwin, S., Text classification using WordNet hypernyms, *In: Proc. Worksh. Usage of WordNet in NLP Systems at COLING-98*, 1998, pp. 38-44.
- [49] Sedding, J., Kazakov, D., WordNet-based textual document clustering, *In: Proc. of COLING-2004 Workshop on Robust Methods in Analysis of Natural Language Data*, 2004.
- [50] Shahnaz, F., Berry, M.W., Pauca, V. P., Plemmons, R. J., Document clustering using nonnegative matrix factorization, *Information Processing and Management*, Vol. 42, No. 2, 2006, pp. 373-386.
- [51] Shihab, K., Improving clustering performance by using feature selection and extraction techniques, *Journal of Intelligent Systems*, Vol. 13, No. 3, 2004, 135-161.
- [52] Srivastava, A. N., Sahami, M., *Text Mining: Classification, Clustering, and Applications*, CRC Press, 2009.
- [53] Steinbach, M., Karypis, G., Kumar, V., A comparison of document clustering techniques, *In: Proc. of the 6th ACM SIGKDD int'l conf. on Knowledge Discovery and Data Mining (KDD)*, 2000.
- [54] Tan, A. H., Text mining: the state of the art and the challenges, *In: Proc. of the Pacific Asia Conf. on Knowledge Discovery and Data mining (PAKDD-99)*, 1999, pp. 65-70.

- [55] Treeratpituk, P., Callan, J., An experimental study on automatically labeling hierarchical clusters using statistical features, *In: Proc. of the 29st Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2006, pp. 707-708.
- [56] Van Rijsbergen, C. J., *Information Retrieval*, second ed., Butterworths, London, 1979.
- [57] Wang, P., Hu, J., Zeng, H. J., Chen, Z., Using wikipedia knowledge to improve text classification, *Knowledge and Information Systems*, Vol. 19, No. 3, 2009, pp. 265-281.
- [58] Wang, Y., Hodges, J., Document clustering with semantic analysis, *In: Proc. of the 39th Annual Hawaii Int'l Conf. on System Sciences*, 2006.
- [59] Wei, C., Hu, P., Dong, Y. X., Managing document categories in E-commerce environments: an evolution-based approach, *European Journal of Information System*, Vol. 11, No. 3, 2002, pp. 208-222.
- [60] Weiss, S. M., Indurkha, N., Zhang, T., Damerau, F. J., *Text Mining: Predictive Methods for Analyzing Unstructured Information*, Springer, Berlin, 2004.
- [61] Willett, P., Recent trends in hierarchic document clustering: a critical review, *Information Processing and Management*, Vol. 24, No. 5, 1988, pp. 577-597.
- [62] Xu, W., Gong, Y., Document clustering by concept factorization, *In: Proc. of the 27th ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2004, pp. 202-209.
- [63] Yu, H., Searsmith, D., Li, X., Han, J., Scalable construction of topic directory with nonparametric closed termset mining, *In: Proc. of ICDM'04*, 2004, pp. 563-566.
- [64] Zadeh, L. A., Fuzzy sets, *Information and Control*, Vol. 8, 1965, pp. 338-353.

- [65] Zhang, C., Wang, H., Liu, Y., Xu, H., Document clustering description extraction and its application, *In: Proc. of the 22nd Int'l Conf. on the Computer Processing of Oriental Languages (ICCPOL2009)*, 2009, pp. 370-377.
- [66] Zhang, W., Yoshida, T., Tang, X., Wang, Q., Text clustering using frequent itemsets, *Knowledge-Based Systems*, Vol. 23, 2010, pp. 379-388.
- [67] Zheng, H. T., Kang, B. Y., Kim, H. G., Exploiting noun phrases and semantic relationships for textual document clustering, *Information Science*, Vol. 179, No. 13, 2009, pp. 2249–2262.
- [68] Zimmermann, H. J., *Fuzzy Set Theory and Its Application*, 2nd Revised Edition, Boston: Kluwer Academic Publisher, 1991.

