TAF Journal of Advances in Technology and Engineering Research JATER
2016, 2(2): 28-34

Document overlapping clustering using formal concept analysis

Yi-Hui Chen [1], Eric Jui-Lin Lu [2,*], Yu-Ting Lin [3], Ya-Wen Cheng [4]

[1] Department of Applied Informatics and Multimedia, Asia University, Taiwan
[2,3,4] Department of Management Information Systems, National Chung Hsing University, Taiwan

Index Terms
Overlapping Clustering
Formal Concept Analysis
Text mining

Received: 19 October 2014
Accepted: 20 July 2015
Published: 26 April 2016

Abstract—Text document clustering is a technique that groups documents into several clusters based on similarities. Most clustering algorithms build disjoint clusters, but clusters should be overlapped because documents may belong to two or more categories in the real world. For example, an article discussing the Apple Watch may be categorized into either 3C, Fashion, or Clothing and Shoes. This paper proposes an overlapping clustering algorithm by using the Formal Concept Analysis, which could make a document assigned to two or more clusters. Moreover, our algorithm reduced the vector space dimensions and performed more efficiently than existing clustering methods.

I. INTRODUCTION

Clustering is a mining technique which groups data (or objects) to clusters by the similarity of data, and is usually implemented in unsupervised learning. Based on the similarity of documents, text document clustering assigns similar documents into a cluster. Because documents in a cluster share the same topic, it could be utilized in categorization or taxonomy of documents.

The previous studies of text document clustering group documents to clusters by using K-means clustering, Hierarchical clustering, etc. But these methods share two common defects. One is that most of them can only assign a document to one cluster. Unfortunately, it is not the case in real world. For example, an article discussing the Apple Watch may be categorized into either 3C, Fashion, or even Clothing and Shoes. The other is that the process of documents transformed to vectors will create too many terms, and thus the size of vectors is tremendously huge.

In this paper, we adopt the Formal Concept Analysis (FCA) to build concept lattices. It will create a hierarchical conceptual structure among documents and keywords. Each node in concept lattice is a formal concept which contains a set of documents and keywords. By properly selecting appropriate concepts and transforming them into clusters, it will assign a document to one or more clusters. Additionally, the size of vectors, which is used to calculate similarity, is reduced. The experimental results on Reuters-21578 collections showed that our methods not only are suitable for producing overlapping clusters efficiently, but also the requirements of memory and computing power are less than existing algorithms.

The remainder of this paper is organized as follows: illustrates the relative work of previous studies, describes the basic formal concept analysis and how to use it in our method, demonstrates the experiment result and uses two metrics to evaluate, and the conclusion and future works.

*Corresponding author: Eric Jui-Lin Lu
E-mail: jllu@nchu.edu.tw

II. LITERATURE REVIEW

Text document clustering is a kind of text mining, the main goal is to discover the important patterns among text documents and group them into clusters. So far many clustering algorithms have been proposed. According to [1], [2] clustering algorithms could be categorized into either hierarchical or partitioning clustering. The most popularly utilized algorithm of hierarchical clustering is Hierarchical Agglomerative clustering (HAC). On the other hand, K-means clustering is the most popular algorithm of partitioning clustering. Most clustering algorithms can only assign a document to no more than one cluster.

However, it is desirable to allow a document to be assigned to more than one cluster in many applications of clustering. As [3] emphasized: It is important to avoid confining each document to only one cluster, so we advocate that clusters should be allowed to overlap. For clustering algorithms that allow two or more clusters share same documents, they are generally called overlapping clustering.

In previous studies, graph-based clustering algorithm is one of common methods to build overlapping clusters. In [3], each document is represented as a node, and the edge between two nodes represents their similarity. After calculating the similarities between all neighbors, core points are selected as the centers of star structure. A cluster is consisted of a core point and its satellites. Because a satellite could, thus, belong to one or more stars, overlapping exists among clusters.

In our previous study [4], we used FCA to cluster blog articles. In experiments, we observed that a very small set of articles were assigned to more than one cluster. In this paper, we attempt to investigate further.

There are other studies which mentioned overlapping clustering [5], [6] and [7], but evaluations of overlapping clustering are limited. Therefore, we aimed to adapt FCA to obtain overlapping clusters and systematically evaluate its result.

III. METHODOLOGY

We use the FCA to assign a single document into formal concepts, convert each formal concept into a concept vector, and cluster the concept vectors using algorithms such as HAC and K-means to build overlapping clusters. The details of the proposed method are described as follows:

A. *Full-Text Segmentation and TF-IDF*

In general, full-text segmentation is the process of extracting terms from documents. Unfortunately, hundreds of terms could be generated and many of them are not really important. To avoid this situation, a value of TF-IDF (term frequency-inverted document frequency) is calculated as the weight (or importance) of each keyword [4]-[8].

B. *Concept Lattice and Concept Vector*

Formal Concept Analysis is a mathematical theory which was proposed by R.Wille in 1980s [9]. The ordered sets and complete lattices are data analysis methods which could transform original data into complete lattices in hierarchical structure [10]. For example, an example document set is shown in Table 1.

TABLE 1
AN EXAMPLE DOCUMENT SET AND THEIR KEYWORDS

Document	Keywords
d_1	Taipei, food, hostel, family, paradise
d_2	Taipei, food, hostel, Taiwan, local dishes
d_3	food, Taiwan, local dishes, hot pot
d_4	hostel, Japan, internet, YouTube
d_5	Internet, YouTube, android, mobile
d_6	Internet, android, mobile, APP
d_7	mobile, iPhone, unboxing, camera
d_8	iPhone, unboxing, design
d_9	mobile, camera, design, pixel, color
d_{10}	mobile, unboxing, color

By using FCA, documents are assigned to concept lattices as shown in Fig. 1. In Fig. 1, we use CLi to represent the set of concepts in the level i of the hierarchy. Based on the design principles of FCA, the concepts in the higher level contain more documents and less keywords. In other words, the concepts in the higher level contain the shared keywords (or semantic meaning) of all its sub-concepts.

As a result, we select all concepts in CL1 in the beginning. Once the concepts are selected, a matrix M is used to transform the concepts into concept vectors. The matrix M is defined as follows:

$$M[i,j] = \begin{cases} 1, & \text{if } d_j \in fc_i \\ 0, & \text{else} \end{cases} \quad (1)$$

Noted that, if the desired number of clusters k is larger than the number of concepts in CL1, CL2 will be selected instead. The process will continue until CLi is greater than k. The matrix M constructed from Fig. 1 is shown in TABLE .

TABLE 2
MATRIX M OF CONCEPT VECTORS FROM Fig. 1

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
fc_1	0	0	0	0	0	0	1	1	0	1
fc_2	0	0	0	0	0	0	0	1	1	0
fc_3	0	0	0	0	1	1	1	0	1	1
fc_4	0	0	0	1	1	1	0	0	0	0
fc_5	1	1	0	1	0	0	0	0	0	0
fc_6	1	1	1	0	0	0	0	0	0	0

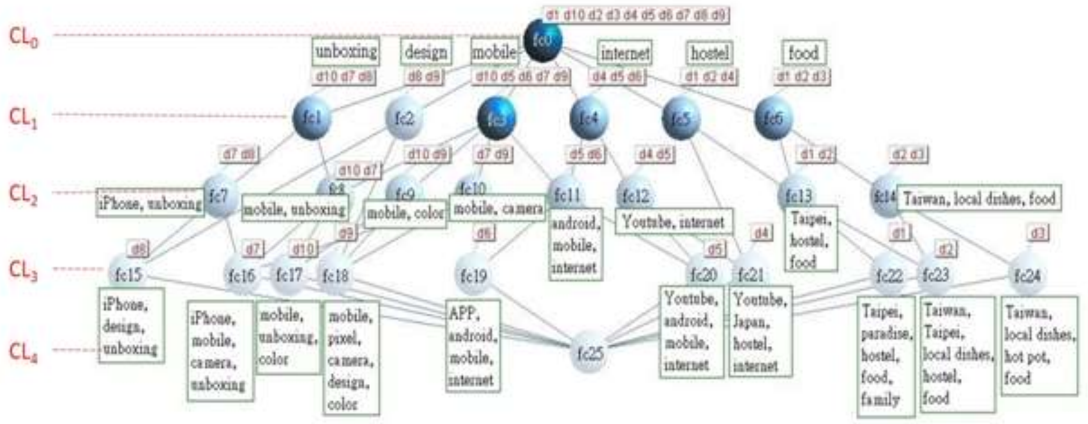

Fig. 1. Concept lattice which is generated by Table 1, we use CL_i to represent the set of concepts in the level i of the hierarchy. The top level of hierarchy is 0, $CL_0 = \{fc_0\}$, and the set of concepts in lower level of CL_0 is $CL_1 = \{fc_1, fc_2, fc_3, fc_4, fc_5, fc_6\}$, and so on.

C. Concept Vectors Clustering

In the beginning, each concept in CLi is treated as a cluster and is represented as a concept vector. As shown in TABLE , six concept vectors are obtained. The concept vector for fc1 is [0, 0, 0, 0, 0, 0, 1, 1, 0, 1]. Once the concept vectors are ready, any cluster algorithm such as K-means or HAC can be used. Because a document can be in more than one concept vector, overlapping clusters can be obtained after clustering.

In this paper, we also revise HAC algorithm proposed by [11]. Instead of using complete link, average link, or single link, we use union operator to combine clusters, and we call it as UHAC (Hierarchical Agglomerative clustering with Union). For example, the most similar formal concepts are fc1 and fc2. After agglomeration, Cnew is [0, 0, 0, 0, 0, 0, 1, 1, 1, 1] which is basically the union of C1 and C2.

D. *Analysis and Evaluation*

In the proposed method, we used concept vectors; instead of the terms extracted from the original documents; to calculate similarity. Thus, the size of vectors is reduced. Supposed that the number of original documents is |D|, and the total number of keywords is |KW|. Without using concept vectors, the size of documents*terms matrix is |D|×|KW|. As describe earlier, the size of matrix M is |D|×|CLi|. In general, |D|×|CLi| is much smaller than |D|×|KW|, especially in a large document corpus. Therefore, the size of vectors is reduced significantly, which means both memory and computing power are saved.

Consider the example in Table 1 there are 10 documents and 20 keywords in total, and the number of concept vectors is 6. Thus, |D|=10, |KW|=20, and |CL1|=6. Without using concept vectors, the size of vector space is 10*20=200. For the matrix M, the size of vector space is 10*6=60 which reduces the vector space by 70%.

To evaluate the performance of the proposed method, the following two metrics are used. The first one is BCubed [12]-[3], [13]. One main advantage of BCubed over other metrics is that there is no need to know the mapping between a cluster and a category in advance. If di and dj share some clusters, the Multiplicity Precision of di and dj is shown in Equation 2. If di and dj share some categories, the Multiplicity Recall of di and dj is shown in Equation 3. In both equations, X(di) is the cluster set of di, and Y(d_i) is the category set of di.

Multiplicity Precision(d_i,d_j)=(Min(|X(d_i)∩X(d_j) |,┤|Y(d_i)∩Y(d_j)|))/|X(d_i)∩X(d_j) | (2)

Multiplicity Recall(d_i,d_j)=(Min(|X(d_i)∩X(d_j) |,┤|Y(d_i)∩Y(d_j)|))/|Y(d_i)∩Y(d_j) | (3)

After calculating the Multiplicity Precision and Multiplicity Recall for each document, the averages of multiplicity precisions and multiplicity recalls, called Precision BCubed and Recall BCubed; respectively, can be used to measure the performance of an overlapping clustering algorithm. Also, [3]proposed FCubed which was extended from F-measures. The formulas of Precision BCubed, Recall BCubed, and FCubed are is shown in Equations 4, 5, and 6; respectively.

Precision BCubed=〚Avg〛_(d_i) [〚Avg〛_(〚 d〛_j ·X(d_i)∩X(d_j)≠∅) [Multiplicity Precision(d_i,d_j)]] (4)

Recall BCubed=〚Avg〛_(d_i) [〚Avg〛_(d_j ·Y(d_i)∩Y(d_j)≠∅) [Multiplicity Recall(d_i,d_j)]] (5)

FCubed=(2×PrecisionBCubed×Recall BCubed)/(Precision BCubed+Recall BCubed) (6)

The other metric is adapted from [14], [15]. The mappings between clusters and categories need to be determined in advance. Supposed that S represents the documents set with pre-defined categories, and D represents the documents set after clustering. In both Equation 7 and 8, X(di) is the cluster set of di, and Y(di) is the category set of di. Also, we revise the traditional F-measure and define F-measure (S, D) as shown in Equation 9.

Precision(S,D)=1/|D| ∑_(i=1)^|D|▒|X(d_i) ∩ Y(d_i)|/|X(d_i) | (7)

Recall(S,D)=1/|D| ∑_(i=1)^|D|▒|X(d_i) ∩ Y(d_i)|/|Y(d_i)| (8)

F-measure (S,D)=(2×Precision(S,D)×Recall(S,D))/(Precision(S,D)+Recall(S,D)) (9)

IV. RESULTS

We use Reuters-21578, which is used widely as document dataset in document clustering. The documents were assigned to categories by personnel from Reuters Ltd. Because a document can be tagged with more than one category, it is also used for overlapping clustering. According to LEWISSPLIT attribute of the REUTERS tag and TOPICS category set, Reuters-21578 was divided into three subsets [13]. The first subset Reu-Te was built from using the LEWISSPLIT attribute tagged as “Test” and associated with at least one topic. The second subset Reu-Tr was built from using the LEWISSPLIT attribute tagged as “Train” and associated with at least one topic. The third subset Reuters is the union of Reu-Te and Reu-Tr.

In the segmentation process, stop words were removed from documents, and words were lemmatized by using the JATE toolkit. After calculating the TF-IDF value [10] of each word, we selected the top 20 words as keywords for each document. We used colibri-java to create the concept lattice of the documents set, and built concept vectors. Then, we used the popular Euclidean distance to calculate the similarities of concept vectors, and experimented with several algorithms to group the concept vectors. The first one is UHAC. The second one is HAC. In the experiments, all three linkages; including Single-linkage, Averagelinkage, and Complete-linkage; were used and denoted as HAC_S, HAC_A, and HAC_C; respectively [16]. The third one is K-means algorithm.

V. CLUSTERING RESULT AND ANALYSIS

As shown in Table 3 the precision BCubed of UHAC in almost all subsets is better than K-means, HAC_S, HAC_A, and HAC_C. However, the recall BCubed of UHAC is worse than other algorithms. As a result, it is recommended to use HAC_A to obtain the best FCubed values. When using the same Reuters-21578 document set, the FCubed values for various overlapping clustering algorithms (Star, ICSD, ACONS, DCS and OClustR) are summarized in Table 4 [13]. It is obvious that the best FCubed value is obtained by using OClustR which is 0.43. However, by using our proposed schemes, the best FCubed value is 0.7875 which is much higher than 0.43.

As shown in
TABLE 5
, precision (S, D), recall (S, D), and F-measure (S, D) were calculated. It is obvious that UHAC outperforms other algorithms. The precision (S, D), the recall (S, D), and F-measure (S, D) of UHAC are at least 3, 7, and 4 times better than the others. Therefore, it is recommended to use UHAC to obtain the best F-measure(S, D).

TABLE 3
THE EXPERIMENTAL RESULTS OF UHAC, K-MEANS, HAC_S, HAC_A, AND HAC_C. THE HIGHEST VALUE PER SUBSET IS DISPLAYED IN BOLD-FACED

Precision BCubed					
Subsets	UHAC	K-means	HAC_S	HAC_A	HAC_C
Reu-Te	0.8021	0.7872	0.8026	0.8026	0.8026
Reu-Tr	0.5275	0.4600	0.4797	0.4797	0.4788
Reuters	0.5404	0.4684	0.4937	0.4937	0.4931
Recall BCubed					
	UHAC	K-means	HAC_S	HAC_A	HAC_C
Reu-Te	0.2276	0.7138	0.7729	0.7729	0.7729
Reu-Tr	0.3201	0.8138	0.8734	0.8739	0.8709
Reuters	0.3213	0.8247	0.8833	0.8838	0.8817
FCubed					
	UHAC	K-means	HAC_S	HAC_A	HAC_C
Reu-Te	0.3546	0.7487	0.7875	0.7875	0.7875
Reu-Tr	0.5878	0.3985	0.6193	0.6194	0.6179
Reuters	0.4030	0.5975	0.6334	0.6335	0.6324

TABLE 4
THE FCUBED VALUES OF STAR, ICSD, ACONS, DCS, AND OCLUSTR. THE HIGHEST VALUE PER SUBSET IS DISPLAYED IN BOLD-FACED

FCubed					
Subsets	Star	ICSD	ACONS	DCS	OClustR
Reu-Te	0.45	0.39	0.40	0.39	0.51
Reu-Tr	0.42	0.36	0.36	0.36	0.43
Reuters	0.42	0.35	0.36	0.35	0.43

ISSN: 2414-4592
DOI: 10.20474/jater-2.2.1

TABLE 5
F-MEASURE (S, D) VALUES IN THE EXPERIMENT. THE HIGHEST VALUE PER SUBSET IS DISPLAYED IN BOLD-FACED

Precision(S, D)					
Subsets	UHAC	K-means	HAC_S	HAC_A	HAC_C
Reu-Te	0.0497	0.0138	0.0103	0.0103	0.0103
Reu-Tr	0.0358	0.0084	0.0071	0.0077	0.0064
Reuters	0.0340	0.0070	0.0070	0.0082	0.0064
Recall(S, D)					
	UHAC	K-means	HAC_S	HAC_A	HAC_C
Reu-Te	0.1624	0.0236	0.0142	0.0142	0.0142
Reu-Tr	0.1631	0.0210	0.0145	0.0150	0.0127
Reuters	0.1616	0.0158	0.0166	0.0156	0.0149
F-measure(S, D)					
	UHAC	K-means	HAC_S	HAC_A	HAC_C
Reu-Te	0.0761	0.0174	0.0119	0.0119	0.0119
Reu-Tr	0.0588	0.0120	0.0095	0.0102	0.0085
Reuters	0.0561	0.0097	0.0099	0.0108	0.0090

VI. DISCUSSION & CONCLUSION

In this paper, we propose an overlapping clustering algorithm that uses the FCA to transform documents into concept vectors. With the proposed method, the requirements of memory and computing power are significantly reduced due to the size of vector space that is also reduced. Additionally, based on the experimental results, our method with FCA design outperforms other previous studies. The proposed UHAC is better than K-means, HAC_S, HAC_A, and HAC_C in precision (S,D), recall (S,D), and F-measure (S,D). For the FCubed metric, HAC-related algorithms are better than K-means.

The Fuzzy clustering [17] is not considered in our method, but it is important to know the “degree” of a document related to clusters. For example, a document may be assigned to both “Basketball” and

“Soccer” clusters. If this document is more related to Basketball, we cannot tell the differences in our method. Therefore, the degree of relatedness of a document to a cluster is worth further investigation [7].

REFERENCES

[1] A. Jain, M. Murty and P. Flynn, "Data clustering: A review. *ACM computing surveys (CSUR)*, vol 31, no. 3, pp. 264-323, 1999.

[2] C. N. Chang, H. R. Ke and W. P. Yang, "*A concept extraction approach for document*," in *International Conference on Advanced Information Technologies*, 2008.

[3] R. Gil-García and A. Pons-Porrata, "Dynamic hierarchical algorithms for document clustering. *Pattern Recognition Letters* vol. 31, no. 6, pp. 469-477, 2010. **DOI:** 10.1016/j.patrec.2009.11.011

[4] Y. H. Chen, E. J. L. Lu and T. Y. Wu, "A blog clustering approach based on queried keywords," in *International Symposium on Biometrics and Security Technologies*, 2013, pp. 1-9. **DOI:** 10.1109/isbast.2013.3

[5] A. Banerjee, C. Krumpelman, S. J. Basu, R. Mooney and J. Ghosh, "Model-based overlapping clustering," in *Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 2005, pp. 532-537. **DOI:** 10.1145/1081870.1081932

[6] F. Bonchi, A. Gionis and A. Ukkonen, "Overlapping correlation clustering," *Knowledge and Information Systems*, vol. 35, no 1, pp. 1-32. **DOI:** 10.1007/s10115-012-0522-9

[7] H. Yu, P. Jiao, G. Wang and Y. Yaoy, "Categorizing overlapping regions in clustering analysis using three-way decisions," in *WI-IAT '14 Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT),* 2014, pp. 350-357. **DOI:** 10.1109/wi-iat.2014.118

[8] C. Luo, Y. Li and S. M. Chung, "Text document clustering based on neighbors," *Data & Knowledge Engineering*, vol 68, no. 11, pp. 1271-1288, 2009. **DOI:** 10.1016/j.datak.2009.06.007

[9] R. Wille, "Formal concept analysis as mathematical theory of concepts and concept hierarchies," in *Formal Concept Analysis*, vol. 3626, New York, NY: Springer Berlin Heidelberg, 2005, pp. 1-33.

[10] J. Gao and W. Lai, "Formal concept analysis based clustering for blog network visualization," in *Lecture Notes in Computer Science*, New York, NY: Springer-Verlag Berlin Heidelberg, 2010, pp. 394-404.

[11] K. A.Heller and Z. Ghahramani, "Bayesian hierarchical clustering," in *Proceedings of the 22nd International Conference on Machine Learning*, 2005. **DOI:** 10.1145/1102351.1102389

[12] E. Amigó, J. Gonzalo, J. Artiles and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," in *Advanced Data Mining and Applications*, Berlin, Germany, Springer Berlin Heidelberg, 2010, pp. 461-486.

[13] A. Pérez-Suárez, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa and J. E. Medina-Pagola, "OClustR: A new graph-based algorithm for overlapping clustering," *Neurocomputing*, vol. 121, no. 9, pp. 234-247, 2013. **DOI:** 10.1016/j.neucom.2013.04.025

[14] S. Godbole and S. Sarawagi, "Discriminative methods for multi-labeled classification," in *Advances in Knowledge Discovery and Data Mining*, vol. 3056, Berlin, Germany: Springer Berlin Heidelberg, 2004, pp. 22-30. **DOI:** 10.1007/978-3-540-24775-3_5

[15] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," in *Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications*, Information Science Reference, 2007, US, pp. 1-13.

[16] C. C. Aggarwal and C. Zhai, A survey of text clustering algorithms," in *Mining Text Data*. New York, NY: Springer, 2012, pp. 77-128.

[17] Y. Yan, L. Chen and W. C. Tjhi, "Fuzzy semi-supervised co-clustering for text documents," *Fuzzy Sets and Systems*, vol. 215, pp. 74-89, 2013. **DOI:** 10.1016/j.fss.2012.10.016

— This article does not have any appendix. —