Text Document Clustering for Topic Discovery by Hypergraph Construction

Wei-San Lin
Graduate Institute of Biomedical Informatic
Taipei Medical University
Taipei, Taiwan 110
Email: g658102003@tmu.edu.tw

Charles Chih-Ho Liu
Cathay General Hospital
Taipei, Taiwan 106
Email: chliu@cgh.org.tw

I-Jen Chiang
Graduate Institute of Biomedical Informatic
Taipei Medical University
Taipei, Taiwan 110
Email: ijchiang@tmu.edu.tw

***Abstract*—The paper presents a hypergraph model and HYPERGRAPH DECOMPOSITION ALGORITHM for text document clustering. The experiments on three different data sets from news, Web, and medical literatures have shown our algorithm is significantly better than traditional clustering algorithms, such as K-MEANS, PRINCIPAL DIRECTION DIVISIVE PARTITIONING , AUTOCLASS and HIERACHICAL CLUSTERING.**

Keywords–document categorization/clustering; hypergraph;association rules; hypergraph components decomposition (HCD); hierarchical clustering (HCA); partition-based hypergraph algorithm.

I. INTRODUCTION

The exponential growth in the volume and the popularity of Web information, such as news, social media, scientific articles and discussion forums, induce a Big Data problem. Since massive amounts of contents have generated everyday, automatically discovering and organizing contextually relevant text information are very challenging.

How to get web sophisticated information mining strategies will be needed. Document clustering can deal with the diverse and large amount of Web information and particularly is used to discover latent concepts in a collection of Web documents, which is inherently useful in organizing, summarizing, disambiguating, and searching through large document collections [1].

Text document clustering is an unsupervised learning technique that has created a demand for a mechanism to discover topics from heterogeneous information. Document clustering aims to generate topic groups or clusters from a document collections. According to a single document, the content can mingle heterogeneous topics, the obtained topics from document clustering methods sometimes do not necessarily correspond to actual topics of interest and document clustering methods do not provide descriptions that summarize the clusters' contents [2]. Many clustering methods, such as k-means, hierarchical clustering (algorithms and non-negative matrix factorization (NMF) have been performed on the matrix to group the documents. However, these methods lack ability of interpretation to each document cluster [3].

In what follows, we start by briefly reviewing the related work in Section II and defining the frequent itemsets in a collection of documents in Section III, and generating a graph model of representing the concepts from the frequent itemsets in Section IV, then presenting the topic based clustering algorithm for partitioning documents into several semantic topics in Section V. In Section VI, you can see each of which represents a concept in the document collection, and documents can then be clustered based on the primitive concepts identified by this algorithm. The three different experimental data sets are also described in Section VII, and finally get into the conclusion in the last section, which showed a novel approach to document clustering which is compared with k-means, HCA, AutoClass or the Principal Direction Divisive Partitioning (PDDP). However, how to provide much more guarantee on precision, even for detailed queries is still an open research problem.

II. RELATED WORK

The frequent itemsets (undirected association rules) can demonstrate semantic topics and can be extracted from documents. A single item, i.e., word, does not carry much information about a document, yet a huge amount of items may nearly identify the document uniquely. Therefore, finding all meaningful frequent itemsets in a collection of textual documents presents a great interest and challenge.

Feldman and his colleagues [4], [5], [6] proposed the *KDT* and *FACT* system to discover association rules based on keywords labelling the documents, the background knowledge of keywords and relationships between them, but it is ineffective. Therefore, an automated approach that documents are labelled by the rules learned from labelled documents [7]. However, several association rules are constructed by a compound word (such as "Wall" and "Street" often co-occur) [8]. Feldman et al. [4], [9] further proposed term extraction modules to generate association rules by selected key words. It is beneficial for us to obtain meaningful results without the need to label documents by human experts. Association rule hypergraph partition was proposed in [10] to transform documents into a transactional database form, and then apply hypergraph partitioning [11] to find the item clusters. Holt and Chung [12] addressed Multipass-Apriori and Multipass-DHP algorithms to efficiently find association rules in text by modified the Apriori algorithm [13] and the DHP algorithm [14] respectively. Those methods did not consider to identify the importance of a word in a document. Hence, they addressed two clustering methods,

 ISBN: 978-1-61208-420-6

CFWS and CFWM, to perform document clustering [15] by considering the sequential aspect of word occurrences.

III. Frequent Itemsets

Association rules was first introduced by Agrawal et al. [16] wherein two standard measures, called *support* and *confidence*, are often used. This paper only focuses on the support; a set of items that meets the support will be called the (undirected) association rules. The association rules are thereby for the use of finding co-occurring frequent terms in documents.

A. Feature Extraction

Feature extraction is to extract key terms from a collection of documents; And various methods such as association rules algorithms may be applied to determine relations between features.

This paper considers only noun entities, especially some representative entities. All NP chunkers extracted by *part-of-speech* (POS) tagger are weighted with respect to the documents after NP chunkers have been recognised and extracted. The simple and sophisticated weighted schema which is most common used in IR or IE is TFIDF indexing, i.e., $\mathrm{tf} \times \mathrm{idf}$ indexing [17], where tf denotes term frequency that appears in the document and idf denotes inverse document frequency [18] where document frequency is the number of documents which contain the NP chunkers. It takes effect on the commonly used NP chunker a relatively small $\mathrm{tf} \times \mathrm{idf}$ value. Moffat and Zobel [19] pointed out that $\mathrm{tf} \times \mathrm{idf}$ function demonstrates: (1) rare NP chunkers are no less important than frequent NP chunkers in according to their idf values; (2) multiple appearances of an NP chunker in a document are no less important than single appearances in according to their tf values. The $\mathrm{tf} \times \mathrm{idf}$ implies the significance of a term in a document, which can be defined as follows.

Definition 1: Let T_r be a collection of documents. The significance of a term, i.e., NP chunker t_i in a document d_j in T_r is its TFIDF value calculated by the function $\mathrm{tfidf}(t_i, d_j)$, which is equivalent to the value $\mathrm{tf}(\mathrm{t_i}, \mathrm{d_j}) \times \mathrm{idf}(\mathrm{t_i}, \mathrm{d_j})$. It can be calculated as

$$\mathrm{tfidf}(\mathrm{t_i}, \mathrm{d_j}) = \mathrm{tf}(\mathrm{t_i}, \mathrm{d_j}) \log \frac{|\mathrm{T_r}|}{|\mathrm{T_r}(\mathrm{t_i})|} \tag{1}$$

where $|T_r(t_i)|$ denotes the number of documents in T_r in which t_i occurs at least once, and

$$\mathrm{tf}(\mathrm{t_i}, \mathrm{d_j}) = \begin{cases} 1 + \log(N(t_i, d_j)) & \text{if } N(t_i, d_j) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $N(t_i, d_j)$ denotes the frequency of terms t_i occurs in document d_j by counting all its nonstop words.

For the purpose of document clustering, we only need to consider when a set of terms that co-occur would become a concept. The metric *support* is used for defining the co-occurred term association. All the documents that are composed of those terms are able to organise a semantic cluster. Let t_A and t_B be two terms. The *support* defined for a collection of documents is as follows.

Definition 2: Support denotes to the specific significance of the documents in T_r that contains both term t_A and term t_B, that is,

$$\mathrm{Support}(\mathrm{t_A}, \mathrm{t_B}) = \frac{\mathrm{tfidf}(\mathrm{t_A}, \mathrm{t_B}, \mathrm{T_r})}{|\mathrm{T_r}|} \tag{3}$$

where

$$\mathrm{tfidf}(\mathrm{t_A}, \mathrm{t_B}, \mathrm{T_r}) = \frac{1}{|\mathrm{T_r}|} \sum_{\mathrm{i}=0}^{|\mathrm{T_r}|} \mathrm{tfidf}(\mathrm{t_A}, \mathrm{t_B}, \mathrm{d_i}) \tag{4}$$

$$\mathrm{tfidf}(\mathrm{t_A}, \mathrm{t_B}, \mathrm{d_i}) = \mathrm{tf}(\mathrm{t_A}, \mathrm{t_B}, \mathrm{d_i}) \log \frac{|\mathrm{T_r}|}{|\mathrm{T_r}(\mathrm{t_A}, \mathrm{t_B})|} \tag{5}$$

and $|T_r(t_A, t_B)|$ define number of documents contained both term t_A and term t_B.

The term frequency $\mathrm{tf}(\mathrm{t_A}, \mathrm{t_B}, \mathrm{d_i})$ of both chunkers t_A and t_B can be calculated as follows.

Definition 3:

$$\mathrm{tf}(\mathrm{t_A}, \mathrm{t_B}, \mathrm{d_j}) = \begin{cases} 1 + \log(\min\{N(t_A, d_j), N(t_B, d_j)\}) \\ \text{if } N(t_A, d_j) > 0 \text{ and } N(t_B, d_j) > 0 \\ 0 \\ \text{otherwise.} \end{cases} \tag{6}$$

A minimal support θ is given to filter the chunkers that their TFIDF values are less than θ. It helps us to eliminate the most common chunkers in a collection and the nonspecific chunkers in a document.

Suppose that with regard to a query term "network", the underlying graph is generated as shown in Figure 1. Each edge denotes the association between two terms is great than a given threshold and illustrates a semantic concept.

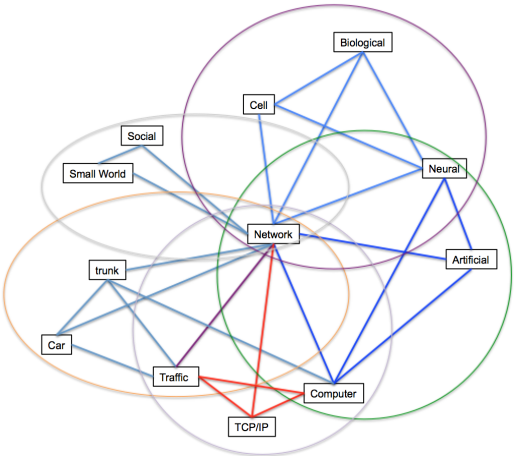

Figure 1. A graph structure generated by the query term "network."

IV. Graph Model of Frequent Itemsets

The set of all frequent itemsets of documents can form a hypergraph of NP chunkers, and this hypergraph can represent the totality of thoughts expressed in this collection of documents. A "simple component" of frequent itemsets organizes hypergraph that represents semantic concepts inside this collection of documents.

 ISBN: 978-1-61208-420-6

A. Preliminary

Let us briefly introduce hypergraphs and define some preliminaries for further descriptions.

Definition 4: A weighted *hypergraph* $G = (V, E, W)$ contains three distinct sets where (1) V is a finite set of vertices, called ground set, (2) $E = \{e_1, e_2, \cdots, e_m\}$ is a non-empty family of finite subsets of V, in which each subset is called a *n-hyperedge* (where $n+1$ is the cardinality of the subset), and $W = \{w_1, w_2, \cdots, w_m\}$ is a weight set. Each hyperedge e_i is assigned a weight w_i.

Two vertices u and v are said to be *r-connected* in a hypergraph if either $u = v$ or there exists a path from u to v (a sequence of r-hyperedge, $(u_j, u_{(j+1)})$, $u_0 = u, \ldots, u_n = v$).

A r-connected hyperedge is called a *r-connected component* or *r-topic*.

B. Concept

For a collection of documents, we generate a hypergraph of frequent itemsets. Note that because of *Apriori* conditions, this hypergraph is closed. The goal of this paper is to establish the following belief.

> Claim A connected component of a hypergraph represents a primitive *concept* in this collection of documents.

Hypergraphs are a perfect method to represent association rules. As seen in Figure 1, the vertex set V ={"network", "artificial", "biological", "car", $\cdots$ } that represents the set of key chunkers in a collection of documents, the edge set E that represents term association rules in the graph. In the graph, each circle represents a higer order association rules, which is a hyperedge. Each circle is also a complete subgraph that its support is bigger than a minimum support, so are all the non-empty subsets of it. In a hypergraph, the universe of vertices organizes 1-item frequent itemsets, the universe of 1-hyperedge represents all possible 1-item and 2-item frequent itemsets, and so on.

V. Topic-based Graph Model

This section will introduce the algorithm to find all frequent itemsets in documents that is generated from the co-occurring chunkers in a collection of documents.

A. Weighted Incident Matrix

The *weighted incident matrix* is

Definition 5:

$$a'_{ij} = \begin{cases} w_{ij} & \text{if } v_i \in e_j \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

where the weight w_{ij} denotes the *support* of an frequent itemset.

Each vertex in V represent a chunker that have been reserved (i.e., its support is greater that a given minimal support θ), and each hyperedge in E is undirected that identifies a support incident with an itemset. Each edge-connector denotes a topic, i.e., an undirected association rules. The number of chunker in an edge-connector defines the *rank* of a hyperedge. An edge-connector of a hyperedge with rank r is said to be a r-hyperedge or r-connected component. As seen in Figure 1, for instance, the set { "network", "artificial", "neural", "computer" } is an edge-connector of a 4-hyperedge that could represents an "artificial neural network" topic.

B. Algorithm

A r-hyperedge denotes a r-topic, which is a r-frequent itemset. If we say a frequent itemset I_i identified by a hyperedge e_i is a subset of a frequent itemset I_j identified by e_j, it means that $e_i \subset e_j$. A hyperedge e_i is said to be a maximal topic if no other hyperedge $e_j \in E$ is the superset of e_i for $i \neq j$. Documents can be automatically clustered based all maximal topics. Considering an example in Figure 1, there are four maximal topics that both of them are 4-hyperedges in a hypergraph. One component is organized by the hyperedge { "network", "artificial", "neural", "computer" }, and { "network", "biological", "neural", "cell" } is another generated hyperedge. The boundary of a concept defines all possible term associations in a document collection. Both of them share a common concept that can be taken as a 1-hyperedge { "network", "neural" }, which is an 2-item frequent itemset. Since all connected components are convex hulls, the intersection of connected components is nothing or a connected component.

Property 1: The intersection of concepts is nothing or a concept that is a maximal closed hyperedge belonging to all intersected concepts.

Since there is at most one maximal closed hyperedge in the intersection of more than one connected topics and the dimension or rank of the intersection is lower than all intersected hyperedges. It is convenient for us to design an efficient algorithm for documents clustering based on all maximal connected components in a hypergraph not needed to traverse all hyperedges. The algorithm for finding all maximal connected components is listed as follows.

Require: $V = \{t_1, t_2, \cdots, t_n\}$ be the vertex set of all reserved NP chunkers in a collection of documents.
Ensure: $\mathcal{E}$ is the set of all maximal connected components.
Let θ be a given minimal support.
$\mathcal{E} \Leftarrow \emptyset$
Let $E_0 = \{e_i | e_i = \{t_i\} \forall t_i \in V\}$ be the 0-hyperedge set.
$i \Leftarrow 0$
while $E_i \neq \emptyset$ **do**
  while for all vertex $t_j \in V$ **do**
    $E_{(i+1)} \Leftarrow \emptyset$ be the $i+1$-hyperedge set.
    while for all element $e \in E_i$ **do**
      if $e' = e \bigcup \{t_j\}$ with $t_j \notin e$ whose *support* is no less than θ **then**
        add e' in $E_{(i+1)}$
        remove e from E_i
      end if
    end while
  end while
  $\mathcal{E} \Leftarrow \mathcal{E} \bigcup E_i$
  $i \Leftarrow i + 1$
end while

All the hyperedges in $\mathcal{E}$ are maximal connected components. A hyperedge will be constructed by including all those co-occurring terms whose support is bigger than or equal to a given minimal support θ. An external vertex will be added into a hyperedge if the produced support is no less than θ. It is not necessary that the intersection of any two hyperedges in $\mathcal{E}$ is empty because the intersection can be taken as the common concept that both own as we have already stated. According to the Property 1, when a maximal connected

 ISBN: 978-1-61208-420-6

component is found, all its subcomponents are also included in the hyperedge.

The documents can be decomposed into several categories based on its correspond concept that is represented by a hyperedge in $\mathcal{E}$. If a document consists in a concept, it means that document highly equates to such concept, thereby all the terms in a concept is also contained in this document. The document can be classified into the category identified with such concept. A document often consists of more than one concept and it can be classified into multi-categories.

VI. Experimental Results

Experimental results are conducted to evaluate the clustering algorithm, rather than analytic statements.

A. Data Sets

Three data sets are involved in making the validation and evaluating the performance of our model and algorithm. Effectiveness is the important criterion for the validity of clustering.

The first dataset is Web pages collected from Boley et al.[10]. 98 Web pages in four broad categories: business and finance, electronic communication and networking, labor and manufacturing are selected for the experiments. Each category is also divided into four subcategories.

The second dataset is the "Reuters-21578, Distribution 1" collection consisted of newswire articles, which is a multi-class, multi-labelled benchmark containing over 21000 newswires articles that are assigned 135 so-called topics. These topics refer to financial news related to different industries, countries and other categories. In our test 9494 documents are selected in which all multi-categorized documents were discarded and the categories with less than five documents have been removed.

The third dataset is 305 electronic medical literatures collected from the journals, *Transfusion*, *Transfusion Medicine*, *Transfusion Science*, *Journal of Pediatrics* and *Archives of Diseases in Childhood Fetal and Neonatal Edition*. Those articles are selected by searching from keywords, *transfusion*, *newborn*, *fetal* and *pediatrics*. The MeSH categories have the use of evaluating the effectiveness of our algorithm. It is best for us to make external validities on the concepts generated from our method by human experts.

B. Evaluation Criteria

The experimental evaluation of document clustering approaches usually measures their *effectiveness* rather than their *efficiency* [20], in the other words, the ability of an approach to make a *right* categorization.

TABLE I. THE CONTINGENCY TABLE FOR CATEGORY c_i.

Category c_i		Clustering Results	
		YES	**NO**
Expert Judgment	**YES**	$\mathrm{TP_i}$	$\mathrm{FN_i}$
	NO	$\mathrm{FP_i}$	$\mathrm{TN_i}$

Considering the contingency table for a category (Table 1), *recall*, *precision*, and F_β are three measures of the effectiveness of a clustering method. Precision and recall with respect to a category is defined as follows:

$$\mathrm{Precision_i} = \frac{\mathrm{TP_i}}{\mathrm{TP_i} + \mathrm{FP_i}} \tag{9}$$

$$\mathrm{Recall_i} = \frac{\mathrm{TP_i}}{\mathrm{TP_i} + \mathrm{FN_i}} \tag{10}$$

The F_β measure combined with precision and recall has introduced by van Rijsbergen in 1979 as the following formula:

$$F_\beta = \frac{(\beta^2 + 1) \times \mathrm{Precision_i} \times \mathrm{Recall_i}}{\beta^2 \times \mathrm{Precision_i} + \mathrm{Recall_i}} \tag{11}$$

F_1 measure is used in this paper, which is obtained when β is set to be 1 that means precision and recall are equally weighted for evaluating the performance of clustering. Because of many categories that will be generated and the comparison reasons, the overall precision and recall are calculated as the average of all precisions and recalls belonging to some categories, respectively. F_1 is calculated as the mean of individual results. It is a macro-average among categories.

In a non-overlapping scenario, each document belongs to exactly one cluster. Three validation metrics: precision, recall and F-measure, are proper to evaluate the performance of crisp clustering algorithms. The overlapping clustering schemes has been involved in a widely variety of application domains because many real problems are naturally overlapped. Information theoretic measures [21], [22], such as entropy and mutual information, hence have been used to estimate how much information is shared from the labelled instances in a cluster, especially, for a hierarchical clustering schemes [23]. In order to compare effectiveness with other methods, two different evaluation metrics, *normalized mutual information* [24], [21], [25] and *overall F-measure* [26], [27], were also used.

C. Results

The result of the first experiment is presented in Table II. The result of PDDP algorithm [10], is under consideration by all non-stop words, that is, the F1 database in their paper, with 16 clusters. The result of our algorithm, HCD, is under consideration by all non-stop words with the minimal support, 0.15 by comparing with four algorithms, HCD, PDDP, k-means and AutoClass. The PDDP algorithm splits the data into two subsets hierarchically. Based on the principal direction, i.e. principal component analysis, it also derives a linear discriminant function. Principal component analysis often hurts the results of classification if with sparse and high dimensional datasets, and induces a high false positive rate and false negative rate. Based on the average of the confidences of the frequent itemsets with the same items, PDDP generates the hyperedges. It is unfair that a possible concept would be withdrawn if a very small confidence of an itemset is existed from an implication direction.

TABLE II. THE PERFORMANCE COMPARISON ON THE FIRST DATASET.

Method	HCD	PDDP	k_means	AutoClass	HCA
Precision	68.3%	65.6%	56.7%	34.2%	35%
Recall	74.2%	68.4%	34.9%	23.6%	22.5%
F_1 measure	0.727	0.67	0.432	0.279	0.274

 ISBN: 978-1-61208-420-6

In the first dataset, HCD generates 47 clusters, i.e., maximal connected components, as shown in Figure 2. It is larger than the original 16 clusters. After performing on decreasing the minimal support value to be 0.1, the number of clusters reduces to be 23 and its precision, recall, and F_1, become 63.7%, 77.3%, 0.698 respectively. The higher the minimal support value is, the lower the number of co-occurred terms in a hypergraph. Precision is worse than PDDP with lower minimal support because the clustering constraints generated from hyperedges are stronger to filter some documents that should be included, which makes a high false positive rate. Figure 3 demonstrates the performance on the first dataset of HCD.

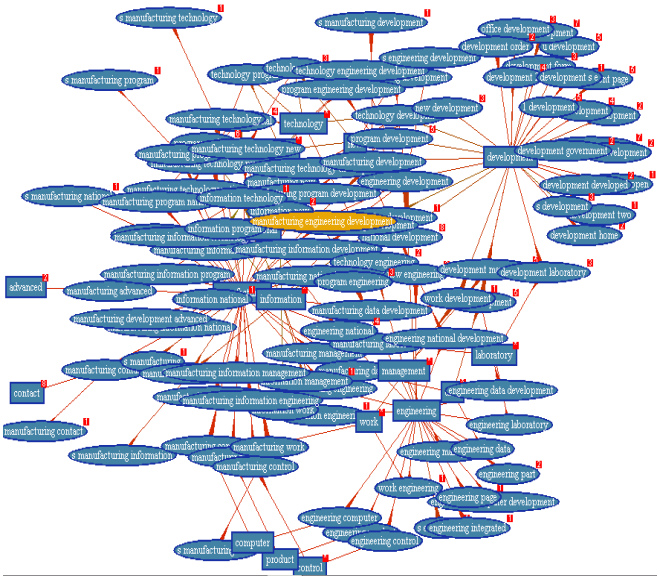

Figure 2. The hypergraph generated from the first dataset by using HCD.

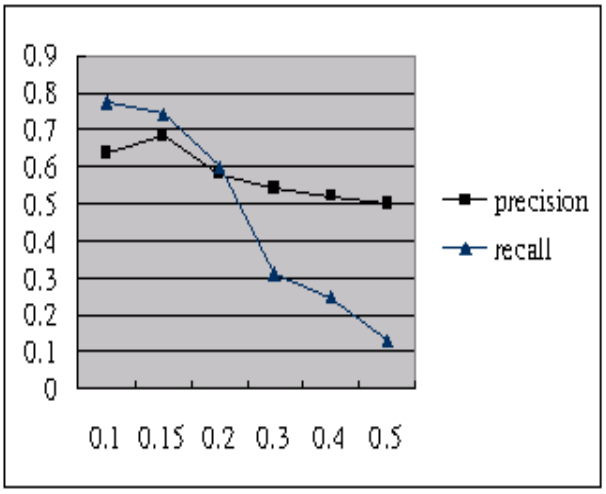

Figure 3. The effectiveness of HCD on the first dataset.

The evaluation was conducted for the cluster numbers ranging from 2 to 10 on the Reuters data set. For each given cluster number k, the performance scores were obtained by averaging those k randomly chosen clusters from the Reuters corpus in an one-run test. Some terms indicated a generic category in Reuters classifications are not designated the same category, so that the number of clusters is larger than the number of Reuters' categories. Table 3 indicates the evaluation results using the Reuters dataset in Figure 4.

TABLE III. THE PERFORMANCE OF REUTERS DATASET BY HCD.

HCD	k=2	k=3	k=4	k=5
Precision	93%	90.8%	93.8%	86.1%
Recall	68%	63.5%	77.9%	76.2%
F_1 measure	0.834	0.774	0.814	0.77

The MeSH categories (22 categories) have been taken to evaluate the effectiveness of HCD on each individual category of the third dataset. Document clustering is based on the MeSH terms related to "Transfusion" and "Pediatrics". The effectiveness of all categories is shown in Figure 5. The MeSH categories are a hierarchical structure that some categories are the subcategories of the other categories. Many concept categories are shared with the same terminology that induces a high false negative rate by HCD on document clustering. In this dataset, documents are not uniform distributed in all categories, some categories only contain a few documents that makes their latent concepts restricted by a few terms, for example, the *Anemia* and the *Surgery* categories whose precision are both below 70%.

VII. CONCLUSION

Concept identification from text documents is an open research problem. While *polysemy*, *phrases* and *chunker dependency* present additional challenges for search technology, single chunker are often insufficient to identify specific concepts in a document. Discriminating NP chunker associations naturally helps distinguish one topic from the others. A group of solid chunker associations can clearly identify a concept. While most methods, like *k-means*, *HCA*, *AutoClass* or *PDDP* classify/cluster documents from the matrix representation, matrix operations cannot discover all chunker associations. Hypergraphs allow a efficient way to find chunker associations in a collection of documents.

This paper presents a novel approach to document clustering based on hypergraph decomposition. An agglomerative method without the use of distance function is proposed. A hypergraph is constructed from the set of co-occurring frequent chunkers in the text documents. The r-hyperedges, i.e., r-topics, can represent basic concepts in the document collection. We presented a simple algorithm that can effectively discover the maximal connected components of co-occurring frequent chunkers. cluster documents. The proposed method is compared with traditional clustering methods, such as *k-means*, *AutoClass* and *HCA*, as well as the partition-based hypergraph algorithm, *PDDP*, on three data sets in our experiments. The hypergraph component decomposition algorithm demonstrated superior performance in document clustering. The results illustrate that hypergraphs are a perfect model to denote association rules in text and is very useful for automatic document clustering.

Our experiments also showed that the value of r is dependent on the given minimal support. The r-connected components represent the r-frequent itemsets with r different

 ISBN: 978-1-61208-420-6

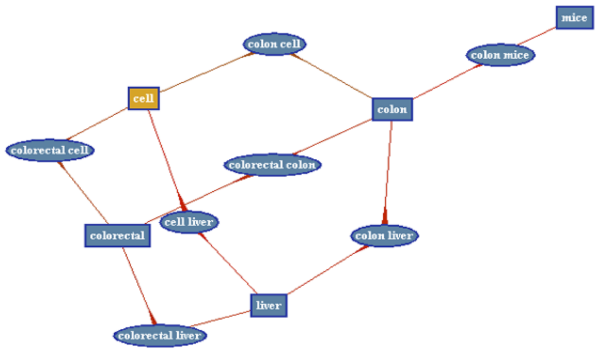

Figure 4. The hypergraph generated from the second dataset with minimal support, 0.1.

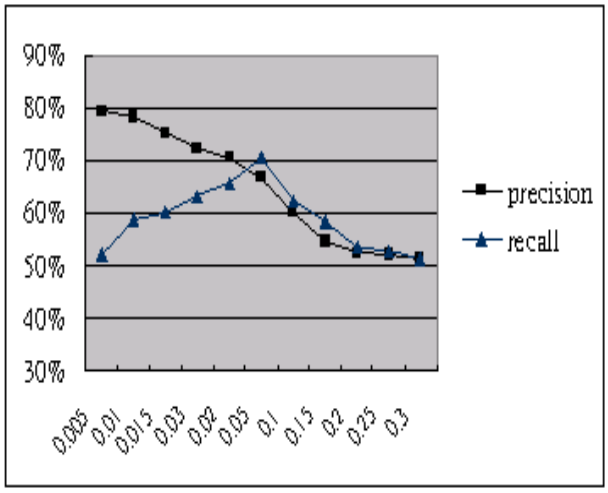

Figure 5. The effectiveness of HCD on the second dataset.

chunkers. The higher the minimal support value is, the lower the value of r is. That is, the number of co-occurring chunkers for organizing concepts in a collection of documents decreases with a higher minimal support value. In other words, the support for a more general concept is higher than the support of a more specific concept. That is, a general concept is less effective in classifying/clustering documents.

The strengths of our methods are: 1) an agglomerative Web document hierarchical clustering is addressed by using graph construction; 2) a hypergraph properly represents the concept organized by the associations of terms in a collection of documents; 3) considering the overlap of semantics between documents, our method can provide more comprehensible clustering results allowing concept overlap. However, as seen in Figure 1, the hyperedge neural, network in the hypergraph is an ambiguous concept. Not until the upper-leveled hyperedges, biological, cell, neural, network and computer, artificial, neural, network have been generated we could clearly identify these two distinct concepts. The weakness of our method is lack of considering uncertainties within documents. We will further consider to develop a fuzzy model on uncertainties.

REFERENCES

[1] R. Kosala and H. Blockeel, "Web mining research: A survey," *SIGKDD Explorations*, vol. 2, no. 1, pp. 1–15, 2000.

[2] H. Anaya-Sánchez, A. Pons-Porrata, and R. Berlanga-Llavori, "A document clustering algorithm for discovering and describing topics," *Pattern Recognition Letters*, vol. 31, pp. 502–510, 2010.

[3] D. Wang, S. Zhu, T. Li, Y. Chi, and Y. Gong, "Integrating document clustering and multidocument summarization," *ACM Transactions on Knowledge Discovery from Data*, vol. 5, no. 3, pp. 14–26, 2011.

[4] R. Feldman, Y. Aumann, A. Amir, W. Klósgen, and A. Zilberstien, "Text mining at the term level," in *Proceedings of 3rd International Conference on Knowledge Discovery, KDD-97*, Newport Beach, CA, 1998, pp. 167–172.

[5] R. Feldman, I. Dagan, and W. Klósgen, "Efficient algorithms for mining and manipulating associations in texts," in *Cybernetics and Systems, The 13th European Meeting on Cybernetics and Research*, vol. II, Vienna, Austria, April 1996.

[6] R. Feldman and H. Hirsh, "Mining associations in text in the presence of background knowledge," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 1996, pp. 343–346.

[7] B. Lent, R. Agrawal, and R. Srikant, "Discovering trends in text databases," in *Proceedings of 3rd International Conference on Knowledge Discovery, KDD-97*, Newport Beach, CA, 1997, pp. 227–230.

[8] M. Rajman and R. Besanon, "Text mining: Natural language techniques and text mining applications," in *Proceedings of seventh IFIP* 2.6 *Working Conference on Database Semantics (DS-7)*, Leysin, Switzerland, 1997.

[9] R. Feldman, M. Fresko, H. Hirsh, Y. Aumann, O. Liphstat, Y. Schler, and M. Rajman, "Knowledge management: A text mining approach," in *Proceedings of 2nd International Conference on Practical Aspects of Knowledge Management*, Basel, Switzerland, 1998, pp. 29–30.

[10] D. Boley, M. Gini, R. Gross, E.-H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore, "Document categorization and query generation on the world wide web using webace," *Artificial Intelligence Review*, vol. 13, no. 5-6, pp. 365–391, 1999.

[11] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partition application in vlsi domain," *Proceedings ACM/IEEE Design Automation Conference*, vol. 8, pp. 381–389, 1997.

[12] J. D. Holt and S. M. Chung, "Efficient mining of association rules in text databases," in *Proceedings of CIKM*, Kansas City, MO, 1999.

[13] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of the 20th International Conference of Very Large Data Bases (VLDB)*, 1994, pp. 487–499.

[14] J. S. Park, M. S. Chen, and P. S. Yu, "Using a hash-based method with transaction trimming for mining association rules," *IEEE Transaction on Knowledge and Data Engineering*, vol. 9, no. 5, pp. 813–825, 1997.

[15] Y. Li, S. M. Chung, and J. D. Holt, "Text document clustering based on frequent word meaning sequences," *Data and Knowledge Engineering*, vol. 64, pp. 381–404, 2008.

[16] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the ACM-SIGMOD 1993 International Conference on Management of Data*, May 1993, pp. 207–216.

[17] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. 513–523, 1960.

[18] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[19] A. Moffat and J. Zobel, "Compression and fast indexing for multi-gigabit text databases," *Australian Computing Journal*, vol. 26, no. 1, p. 19, 1994.

[20] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, pp. 1–47, 2002.

[21] M. Meilă, "Comparing clusterings-an information based distance," *Journal of Multivariate Analysis*, vol. 98, pp. 873–895, 2007.

[22] M. Sokolova and G. Lapalme, "A systematic analysis of performance

measures for classification tasks," *Information Processing and Management*, vol. 45, pp. 427–437, 2009.

[23] M. Aghagolzadeh, H. Soltanian-Zadeh, and B. N. Araabi, "Information theoretic hierarchical clustering," *Entropy*, vol. 13, pp. 450–465, 2011.

[24] T. Cao, H. Do, D. Hong, and T. Quan, "Fuzzy named entity-based document clustering," in *Proc. of the 17th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2008)*, Hong Kong, 2008, pp. 2028–2034.

[25] W. Xu and Y. Gong, "Document clustering by concept factorization," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, Sheffield, United Kingdom, 2004, pp. 202–209.

[26] A. Dalli, "Adaptation of the f-measure to cluster based lexicon quality evaluation," in *Proceedings of the EACL 2003 Workshop on Evaluation Initiatives in Natural Language Processing: are evaluation methods, metrics and resources reusable?*, Budapest, Hungary, 2003, pp. 51–56.

[27] K. Kummamuru, R. Lotlikar, S. Roy, K. Singal, and R. Krishnapuram, "A hierarchical monothetic document clustering algorithm for summarization and browsing search results," in *Proceedings of the 13th international conference on World Wide Web*, New York, NY, 2004, pp. 658–665.

 ISBN: 978-1-61208-420-6