GENERATING HYPERGRAPH OF TERM ASSOCIATIONS FOR AUTOMATIC DOCUMENT CONCEPT CLUSTERING

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

Tsau Young ('T. Y.') Lin
Department of Computer Science
San Jose State University
208 MacQuarrie Hall, College of Science
San Jose, CA 95192-0249
email: tylin@cs.sjsu.edu

Jane Yung-jen Hsu
Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan 100
email: yjhsu@csie.ntu.edu.tw

Abstract

This paper presents a novel approach to document clustering using hypergraph decomposition. Given a set of documents, the associations among frequently co-occurring terms in any of the documents define naturally a hypergraph, which can then be decomposed into connected components at various levels. Each connected component represents a primitive concept in the collection. The documents can then be clustered based on the primitive concepts. Experiments with three different data sets from web pages and medical literatures have shown that the proposed unsupervised clustering approach performs significantly better than traditional clustering algorithms, such as *k-means*, *AutoClass* and *Hierarchical Clustering* (HAC). The results indicate that hypergraphs are a perfect model to capture association rules in text and is very useful for automatic document clustering.

KEY WORDS
Document Clustering, Hypergraph, Association Rules, Concept, Connected Components, Decomposition

1 Introduction

Due to the rapid growth of resources over the Web and the diversity of content within any web page, automatic tools are necessary to help users find, filter, and extract the desired information. Search engines have become indispensable tools for gathering web pages and documents that are relevant to a user's query. Unfortunately, inconsistent, uninteresting and disorganized search results are often returned. Without conceptual contexts, issues like *polysemy*, *phrases* and *term dependency* impose limitations on search technology [8]. Search results can be imporved with mechanisms based on categories, subjects, and contents.

Document clustering is considered as a mechanism to improve search restuls. A good search engine needs to discriminate whether a piece of information is relevant to users' queries within a short time. Short of the ability to extract semantic meaning from a document automatically, one hopes to find a technique that can classify or cluster Web documents into semantic categories based on extracted features from those documents. Given that multiple concepts can be simultaneously defined in a single Web page, it is hard to limit the number of concept categories in a collection of Web pages. As a result, unsupervised clustering methods are better suited for document categorization on the huge, diverse, and scattered Web.

Our most important observation is that the frequent itemsets (undirected association rules) that can be identified in a collection of documents naturally form a *hypergraph* [11]. In this research, we explore whether hypergraph partitioning represents a significant improvement over the traditional methods, such as *k-means*, *AutoClass* [5] and *Hierarchical Clustering* (HAC) algorithms. Boley et al. [4] proposed a partition-based hypergraph algorithm, PDDP, to hierarchically split data into two branches, which are two hyperedges based on the principal direction. The average of the confidences of the itemsets is to determine the hyperedges being generated or not. It is unfair if a very small confidence of an itemset is existed from an implication direction. This paper proposes a bottom-up hypergraph decomposition algorithm based on the support of itemsets that is able to solve the problem.

In what follows, we start by reviewing related work on Web document clustering in section 2. Section 3 defines the association rules in a collection of documents and illustrates the way to compute the *support* and *confidence* of each association rule. The concept and construction of hypergraphs from the frequent itemsets generated by association rules is given in section 4. Section 5 presents the hypergraph clustering algorithm for partitioning a hypergraph into several subhypergraphs, each of which represents a concept in the document collection. Documents can then be clustered based on the primitive concepts identified

by this algorithm. Experimental results from three different data sets are described in Section 6; followed by the conclusion.

2 Related Work

Most search engines provide instant gratification in response to user queries, however, they provide little guarantee on precision, even for detailed queries. There has been much research on developing more intelligent tools for information retrieval, such as machine learning [14], text mining, and intelligent Web agents [12].

Document clustering has been considered as one of the most crucial techniques for dealing with the diverse and large amount of information present on the World Wide Web. In particular, clustering 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 [9].

Many methods, including *k-means*, hierarchical clustering and nearest-neighbor clustering etc., select a set of key terms or phrases to organize the feature vectors corresponding to different documents. Suffix-tree clustering [15], a phrase-based approach, formed document clusters depending on the similarity between documents.

When the number of features selected from each document is too large, methods for extracting the salient features are taken. However, the residual dimension can still be very large, and the quality of the resulting clusters tends to be not as good due to the loss of relevant features. Frameworks for reducing the dimension of the feature space include principle component analysis, independent component analysis, and latent semantic indexing [1, 3]. Furthermore, in the presence of noise in the data, feature extraction may result in degradation of clustering quality [4]. Association rule hypergraph partition was first proposed in [4] to transform documents into a transactional database form, and then apply hypergraph partitioning to find the item clusters.

Cutting et al. introduced partition-based clustering algorithms document clustering [6]. Buckshot and fractionation were developed in [10]. Greedy heuristic methods are used in the hierarchical frequent term-based clustering algorithm [2] to perform hierarchical document clustering by using frequent itemsets.

3 Keywords, Associations, and Documents

The word or phrase frequency distribution in a document collection is quite different from the item frequency distribution in a retail sales transaction database. Documents are amorphous. A single word does not carry much information about a document, yet a huge amount of words may nearly identify the document uniquely. So finding all association rules in a collection of textual documents presents a great interest and challenge.

Feldman and his colleagues [7] proposed the *KDT* and *FACT* system to discover association rules based on keywords labeling the documents, the background knowledge of keywords and relationships between them. This is ineffective because a substantially large amount of background knowledge is required. Therefore, the use of term extraction modules have been propose to generate association rules by selected key words [7]. It is beneficial for us to obtain meaningful results without the need to label documents by human experts.

The TFIDF value is the weight of term in each document. While considering relevant documents to a search query, if the TFIDF value of a term is large, then it will pull more weight than terms with lesser TFIDF values.

Regarding to TFIDF values, *information extraction* is taken to rank words in a document, which are able to identify the terms (features) that are expected to yield the best effectiveness from a set of documents.

3.1 Feature Extraction

A general framework for text mining is consisted of two phrases. The first step *feature extraction* is to extract key terms from a collection of "indexed" documents; as a second step various methods such as association rules algorithms may be applied to determine relations between features.

The most simple and sophisticated weighted schema which is most common used in information retrieval or information extraction is TFIDF indexing, i.e., $\mathrm{tf} \times \mathrm{idf}$ indexing, where tf denotes term frequency that appears in the document and idf denotes inverse document frequency where document frequency is the number of documents which contain the term. It takes effect on the commonly used word a relatively small $\mathrm{tf} \times \mathrm{idf}$ value. Moffat and Zobel [13] pointed out that $\mathrm{tf} \times \mathrm{idf}$ function demonstrates: (1) rare terms are no less important than frequent terms in according to their idf values; (2) multiple appearances of a term 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 denote a collection of documents. The significance of a term t_i in a document d_j in T_r is its TFIDF value calculated by the function* $\mathrm{tfidf}(\mathrm{t_i}, \mathrm{d_j})$*, 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*

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

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

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

where $N(t_i, d_j)$ denotes the frequency of terms t_i occurs in document d_j.

For the document clustering purpose, we throw the redundant definition of *confidence* and propose a simple idea on undirected association rules.

3.2 Undirected Term Associations

Support and *confidence* are in use for defining association rules in a transaction database. For the purpose of document clustering, we only need to consider when a set of terms that co-occur would become a concept. All the documents that are composed of those terms are able to organize a semantic cluster. The *confidence* is unnecessary in our framework. Undirected association rules are determined only by the *support*, which is defined in this subsection for a document collection. 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,*

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

where

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

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

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 term t_A and t_B can be calculated as follows.

Definition 3

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

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

4 Formal Theory of Hypergraph

First we observe that the set of all association rules for a collection of documents, naturally, forms a hypergraph of key terms. We believe this hypergraph captures the totality of thoughts expressed in this collection of documents; and a "simple component" (which is a *r-connected component*) of the hypergraph represents some one primitive unit of concept inside this collection.

4.1 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 nonempty 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. If all weights are the same, then we say the weighted hypergraph is unweighted.*

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*.

4.2 Main Idea

For a collection of documents, we generate a hypergraph of association rules. 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.

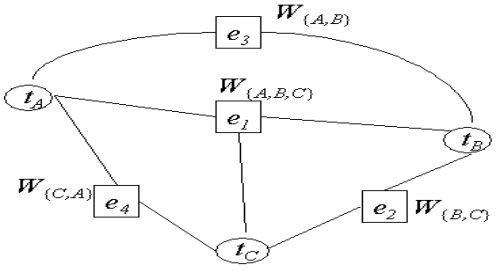

Figure 1. A sample hypergraph example.

A hypergraph example is depicted in Figure 1. In the graph, the vertex set $V = \{t_A, t_B, t_C\}$ that represents the set of three key terms in a collection of documents, the edge set $E = \{e_1, e_2, e_3, e_4\}$ that represents association rules in V, and $W = \{w_{A,B}, w_{C,A}, w_{B,C}, w_{A,B,C}\}$ in which each weigh denotes the support on an association rule.

This property satisfies the criterion of association rules: if the support of an item set $\{t_1, t_2, \cdots, t_n\}$ is bigger that a minimum support, so are all the nonempty subsets of it. Hypergraphs are a perfect method to represent association rules. 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.

5 Hypergraph Components Decompositions

This section will introduce the hypergraph components decomposition (HCD) algorithm to find all concepts, i.e., connection components, in a hypergraph that is generated from the found co-occurring terms in a collection of documents.

5.1 Hypergraph Presentation

In order for the further discussion on the hypergraph components decomposition, let us make the following definitions.

The *incident matrix* and the *weighted incident matrix* are defined as follows.

Definition 5 *The* $n \times m$ incident matrix $A = (a_{ij})$ *associated to a hypergraph is defined as*

$$a_{ij} = \begin{cases} 1 & \textit{if } v_i \in e_j \\ 0 & \textit{otherwise}. \end{cases}$$

The corresponding weighted incident matrix $A' = (a'_{ij})$ *is*

$$a'_{ij} = \begin{cases} w_{ij} & \textit{if } v_i \in e_j \\ 0 & \textit{otherwise}. \end{cases},$$

where the weight w_{ij} *denotes the* support *of an association rule.*

Each vertex in V represent a term 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 connected component, i.e., an undirected association rules. The number of terms 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, the edge-connector of a 3-hyperedge e_1 is the set $\{t_A, t_B, t_C\}$, which is a connected component that represents a concept in a document collection.

5.2 Algorithm

A r-hyperedge denotes a r-connected component, 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 connected component 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 connected components.

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 components and the dimension or rank of the intersection is lower than all intersected hyperedges. An efficient algorithm for documents clustering based on all maximal connected components in a hypergraph not needed to traverse all hyperedges is easy obtained as follows.

Require: $V = \{t_1, t_2, \cdots, t_n\}$ be the vertex set of all reserved terms 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
  $\mathcal{E} \Leftarrow \mathcal{E} \bigcup \mathrm{E}_i$
  $i \Leftarrow i + 1$
end while

The documents can be decomposed into several categories based on its correspond concept that is represented by a hyperedge in $\mathcal{E}$. All the hyperedges in $\mathcal{E}$ are maximal connected components 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. According to the Property 1, when a maximal connected component is found, all its subcomponents are also included in the hyperedge.

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.

6 Experimental Results

As for text search systems and document categorization systems, experimental results are conducted to evaluate the clustering algorithm, rather than analytic statements.

6.1 Data Sets

Three kinds of datasets are experiments are taken in our study. The first dataset is Web pages collected from Boley

et al.[4]: 98 Web pages in four broad categories and each category is also divided into four subcategories.

The second dataset is 848 electronic medical literature abstracts collected from *PubMed*. All those abstracts are collected by searching from the keywords of *cancer*, *metastasis*, *gene* and *colon*. Our purpose is to discriminate all articles in according to which organs a cancer spreads from the primary tumor. In our study, we neglect the primary tumor is occurred in colon or from the other organs. A few organs are selected for this study, such as, liver, breast, lung, brain, prostate, stomach, pancreas, and lymph.

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.

6.2 Results

The experimental evaluation of document clustering approaches usually measures their *effectiveness* rather than their *efficiency* [14], in the other word, the ability of an approach to make a *right* categorization. *Recall*, *precision*, and F are three measures of the effectiveness of a clustering method.

Table 2 demonstrates the results of the first experiment. The result of the algorithm, PDDP [4], 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. The PDDP algorithm hierarchically

Table 1. The first dataset is compared with four algorithms, HCD, PDDP, k-means and AutoClass.

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

splits the data into two subsets, and derives a linear discriminant function from them based on the principal direction (i.e., principal component analysis). With sparse and high dimensional datasets, principal component analyses often hurt the results of classification, which induces a high false positive rate and false negative rate.

The effectiveness of the second dataset is shown in Figure 3. The use of fourteen organ related words are selected for clustering those abstracts. Figure 5 demonstrates the generated hypergraph associated with a minimal support, 0.05.

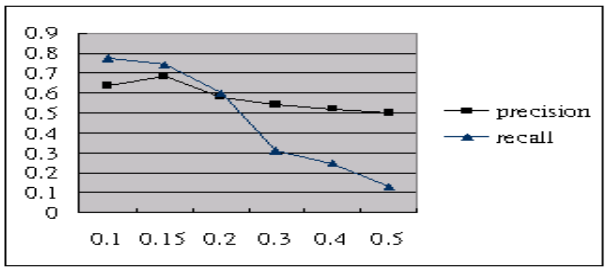

Figure 2. The effectiveness of HCD on the first dataset.

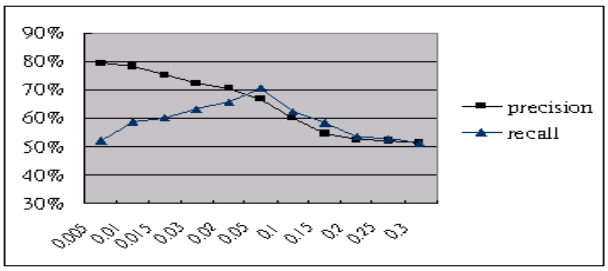

Figure 3. The effectiveness of HCD on the second dataset.

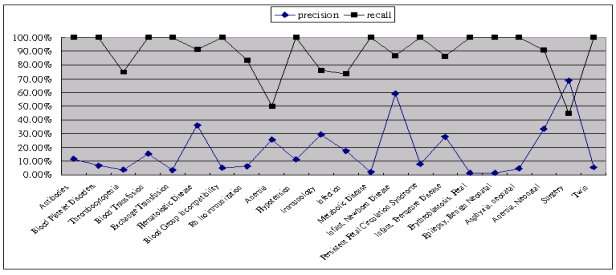

Figure 4. The effectiveness of HCD of the third experiment with minimal support, 0.02.

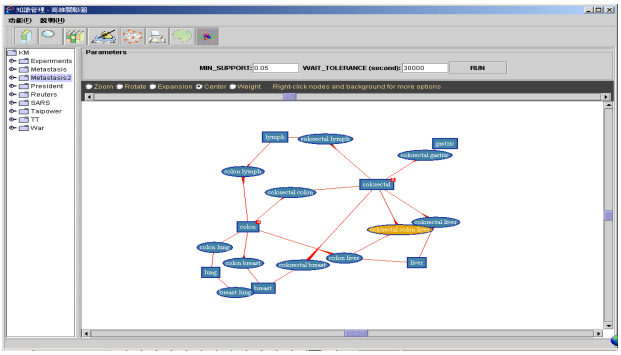

Figure 5. The hypergraph generated from the second dataset with minimal support, 0.05.

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 4. 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 terminologies 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%.

7 Conclusion

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

The paper presents a noval 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 terms in the text documents. The r-hyperedges, i.e., r-connected components, can represent basic concepts in the document collection. Comparing with traditional clustering methods, such as *k-means*, *AutoClass* and *Hierarchical Clustering* (HAC), as well as the partition-based hypergraph algorithm, *PDDP*, on three data sets, 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.

References

[1] T. W. Anderson. On estimation of parameters in latent structure analysis. *Psychometrika*, 19:1–10, 1954.

[2] F. Beil, M. Ester, and X. Xu. Frequent term-based text clustering. In *Proceedings of 8th Int. Conf. on Knowledge Discovery and Data Mining (KDD 2002)*, Edmonton, Alberta, Canada, 2002.

[3] M. W. Berry. Large scale sparse singular value computations. *International Journal of Supercomputer Applications*, 6(1):13–49, 1992.

[4] 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*, 13(5-6):365–391, 1999.

[5] P. Cheeseman and J. Stutz. Baysian classification (autoclass): Theory and results. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smith, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 153–180. AAAI/MIT Press, 1996.

[6] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference*, pages 318–329, 1992.

[7] 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*, pages 167–172, Newport Beach, CA, 1998.

[8] A. Joshi and Z. Jiang. Retriever: Improving web search engine results using clustering. In A. Gangopadhyay, editor, *Managing Business with Electronic Commerce: Issues and Trends*, chapter 4. World Scientific, 2001.

[9] R. Kosala and H. Blockeel. Web mining research: A survey. *SIGKDD Explorations*, 2(1):1–15, 2000.

[10] K. I. Lin and H. Chen. Automatic information discovery from the invisible web. In *Proceedings of the The International Conference on Information Technology: Coding and Computing (ITCC'02), Special Session on Web and Hypermedia Systems*, 2002.

[11] T. Y. Lin and I. J. Chiang. Automatic document clustering of concept hypergraph decompositions. In *Proceedings of SPIE*, volume 5098, pages 168–177, Orlando, FL, 2004.

[12] D. Mladenic. Text-learning and related intelligent agents: A survey. *IEEE Intelligent Systems*, pages 44–54, 1999.

[13] A. Moffat and J. Zobel. Compression and fast indexing for multi-gigabit text databases. *Australian Computing Journal*, 26(1):19, 1994.

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

[15] O. Zamir and O. Etzioni. Web document clustering: a feasibility demonstration. In *Proceedings of 19th international ACM SIGIR conference on research and development in information retrieval (SIGIR 98)*, pages 46–54, 1998.