A Semi-Supervised Document Clustering Technique for Information Organization

Han-joon Kim and Sang-goo Lee
School of Computer Science and Engineering
Seoul National University
San 56-1, Shillim-dong, Kwanak-gu, Seoul, Korea, 151-742
{hjkim,sglee}@europa.snu.ac.kr

ABSTRACT
This paper discusses a new type of semi-supervised document clustering that uses partial supervision to partition a large set of documents. Most clustering methods organize documents into groups based only on similarity measures. Unfortunately, the traditional approaches to document clustering are often unable to correctly discern structural details hidden within the document corpus because their algorithms inherently strongly depend on the document themselves and their similarity to each other. In this paper, we attempt to isolate more semantically coherent clusters by employing the domain-specific knowledge provided by a document analyst. By using external human knowledge to guide the clustering mechanism with some flexibility when creating the clusters, clustering efficiency can be considerably enhanced. As a basic clustering strategy, we use a variant of complete-linkage agglomerative hierarchical clustering, and develop the concepts (or seeds) of requested clusters by exploiting user-relevance feedback. Although the proposed method is slow when applied to large document collection, it yields higher quality clusters. Through experiments using the Reuters-21578 corpus, we show that the proposed method outperforms unsupervised clustering method.

Keywords
Document Clustering, Agglomerative Hierarchical Clustering, Information Organization, Relevance Feedback, Fuzzy Information Retrieval

1. INTRODUCTION
The goal of document clustering is to organize a large document collection (corpus) into groups of documents that are related among themselves, and to discern the most common general themes hidden within the corpus. In modern information systems, document clustering is essential in dealing with large collections and is an integral part of information retrieval. Since clusters are distinct groups of similar documents, they can be thought of as representing topically coherent subtopics in the corpus. In tackling modern information overload problem, organizing documents according to their subtopics is of importance to both information organization and information retrieval, although the two aspects cannot be clearly separated.

Recent information systems have used different clustering algorithms to hierarchically organize a large document set according to topic [?][?] [?]. If a document collection is continuously updated, then it must be possible for new topics to be found and relevant documents isolated, which is easier with some clustering techniques. Also, in an environment where documents are distributed on several information servers, relevant documents need to be clustered for query routing, which is a process of directing user queries to appropriate servers. In such systems, documents that are stored close together on the same server can be retrieved with minimal disk head movement. Secondly, document clustering has been accepted as a good methodology for enhancing conventional document search and retrieval. The underlying reason for this enhancement is referred to as the ‘cluster hypothesis’, which states that relevant documents tend to be more closely related to each other than to other non-relevant documents [?]. Queries are matched against clusters and the contents of the highest scoring clusters are returned as the results. Clustering can also be used to drill down to the result of a query. In [?] and [?], such cluster-based browsing was evaluated as very efficient in improving retrieval results. In this paper, our goal is related to the first aspect, that of building a well-organized information system through clustering methods.

Unfortunately, traditional approaches to document clustering are not intended to correctly discern structural information hidden within a document collection. Data-driven clustering algorithms cannot solve the ‘word mismatch problem’ that occurs when different words are used to describe the same concept, since the algorithms used inherently strongly depend on the documents themselves and on a similarity measure. Moreover, clustering allows more freedom in the number of clusters created and the kind of document assigned to each cluster. Unlike classification tasks, where user knowledge is provided, most unsupervised clustering methods merely attempt to optimize an internal measure

*This work was supported by the Brain Korea 21 project.
of clustering quality, such as the overall similarity within clusters.

Clustering is a task that may significantly benefit from the incorporation of prior or external knowledge. The incorporated knowledge reflects specific user goals or reasoning paths, and may contribute to better results. It should be possible to organize information in a way that more closely reflects the interests of users than the fixed viewpoints provided by conventional clustering methods. Such a supervised clustering technique is particularly necessary when the clusters that result with previous methods are not compact and well-separated groups.

We are trying to develop a kind of semi-supervised clustering technique that has the capability to use external knowledge provided by the user during the clustering process. Our goal is to partition a given document collection into disjoint clusters, where each cluster is as consistent as possible in terms of relevance information provided by the user interview. As we are considering document clustering to organize information, the role in the clustering process of prior knowledge can be considered highly related to user interaction. As a basic clustering strategy, we adopt the classical agglomerative hierarchical clustering algorithm. To enhance the clustering efficiency, we introduce a concept of ‘fixed-diameter’ clusters, where judgment of a document’s relevance to others depends only on distance. Also, when obtaining user knowledge, we adopt the relevance-feedback learning technique that researchers in information retrieval have advocated for information retrieval.

2. RELATED WORK
Several researchers have considered how best to incorporate external knowledge into a clustering system. A related approach is the ‘user-oriented clustering’ proposal, where clusters are identified on the basis of relevance feedback, without depending on an index term [7] [8] [9]. This approach requires that the system accumulate a long-term query information result for sets of past queries. To identify clusters, it has to solve the boundary selection problem, which is known to be NP-hard. Furthermore, this type of user-oriented clustering can encounter difficulties when documents are found to be relevant for different queries. [7] proposed a way to provide declarative external knowledge, a set of classification rules expressed in first-order logic, to a hierarchical clustering algorithm. However, this approach is unsuitable for text data that has a large number of attributes (index terms), since it is nearly impossible for users to express sufficient classification knowledge of such index terms as to help the clustering process. In [9], the representation of documents within a cluster is re-described by using a genetic algorithm, so that the representation of a cluster resembles the representation of its relevant queries.

Another approach is a semi-supervised clustering, which combines the benefits of supervised and unsupervised learning methods [7] [8] [9]. [7] is analogous to our approach. However, it basically uses the classical k-means clustering method, which has a number of limitations that are incompatible with document clustering. The number of clusters must be specified prior to the clustering process, and the method breaks down as the number of clusters increases [9]. In [8] and [9], prior knowledge is extracted from categories in an existing hand-built hierarchical topical tree, and used to develop seed points for the construction of clusters. Still, such limited supervision is not enough to generate accurate clusters that are adaptable to individual users.

3. PRELIMINARIES
3.1 Document Representation
As in standard information retrieval systems, we use the vector space model to represent documents as points in a high-dimensional topic space, where each dimension corresponds to a unique word from the corpus. In the vector space model, the semantics of documents are expressed through a set of index terms. Therefore, each document can be represented as a vector of the form \( \vec{d}_i = (w_{i1}, w_{i2}, \ldots, w_{in}) \) where \( n \) is the total number of index terms in the system, and \( w_{ij} \) denotes the weighted frequency at which term \( j \) occurs in document \( i \). We use a normalized \( \text{tf} \) weighting scheme where the size of a document vector is scaled to one. In an un-normalized \( \text{tf} \) weighting scheme, a longer document may contain components that cause it to become too far from a given document when computing the distance between vectors. We do not use the commonly used \( \text{tf-idf} \) weighting scheme, because it is inefficient in a clustering algorithm [7]. Although the \( \text{idf} \) weighting scheme may be useful for a similarity search, it hinders correct clustering, since clusters tend to be formed that are based on patterns of frequently used words.

The similarity between two documents can be measured by calculating the cosine value between the document vectors. Because each document vector is normalized (i.e., all the documents can be represented on the multi-dimensional circle of unit size), the cosine value is equated to an inner product of two document vectors. Moreover, this cosine similarity can be used interchangeably with the Euclidean distance.

3.2 Cluster Representation
While the classical centroid-based clustering methods such as k-means clustering assumes that clusters are hyper-ellipsoidal, our approach that incorporates a user’s knowledge can find clusters of arbitrary shape, in which hyper-ellipsoidal sub-clusters are distributed. Therefore, a cluster should be represented as a set of multiple representative prototypes. Each prototype vector can be obtained by averaging the vectors of the documents nearest to that prototype. We call each prototype within a cluster a ‘local prototype’.

In order to produce more accurate and precise clusters that reflect hidden topic structures, we attempt to generate tighter and more finely grained clusters, in which relevance can be determined by the distance between any two documents. The cluster hypothesis holds more strongly when the value for the diameter of a cluster is below a small threshold. In our proposed clustering algorithm, small ‘fixed-diameter’ clusters are generated first. The diameter of a cluster is defined in the following.

Given \( m \) n-dimensional data points in a cluster \( c = \{\vec{d}_1, \ldots, \vec{d}_m\} \),
the diameter \( \delta \) of a cluster is defined in the two formulae:

\[
\delta = \max_{\vec{d}_i, \vec{d}_j \in c} |\vec{d}_i - \vec{d}_j| \quad (1)
\]

\[
\delta = \sqrt{\frac{\sum_{i=1}^{m} \sum_{j=1}^{m} (\vec{d}_i - \vec{d}_j)^2}{m(m-1)}} \quad (2)
\]

In formula (1), the diameter \( \delta \) of a cluster is defined as the maximum distance between two document points within the cluster. As seen in formula (2), \( \delta \) can also be defined as the average pair wise distance within a cluster. These two measurements are two alternative measures of the tightness of a cluster around a centroid. Since we use a normalized \( tf \) weighting scheme, the values range between 0 and \( \sqrt{2} \).

### 3.3 Classical Agglomerative Hierarchical Clustering

Agglomerative hierarchical clustering (AHC) algorithms are most commonly used in document clustering. These algorithms start with each document as a separate cluster, and then successively merge the two most similar clusters until a termination criterion is achieved, such as when the number of resulting clusters reduces to a given value. Therefore, the clustering algorithms generate hierarchy trees of clusters (also called dendograms) whose leaves are individual points; the internal nodes correspond to clusters formed by clusters merging at their children.

Since our focus is clustering methods for information (re-)organization and topic discovery, precise clustering is of more interest than fast clustering. Thus, to produce fixed-diameter clusters, we use a variant of complete-linkage clustering, which is one of various agglomerative hierarchical algorithms. In this method, the similarity between two clusters is defined as the minimum of the similarities between all pairs of inter-cluster documents. As a result of this minimal-ity criterion, the resulting clusters tend to be small and tight when the documents are all very similar. Although complete linkage algorithms are typically slow when applied to large document collections (typically requires \( O(n^2) \) time), in terms of quality, in comparative studies of document retrieval the complete-link algorithm has been shown to perform well [?].

### 3.4 Relevance Feedback

For supervised clustering, we allow users to define a set of document groups that is considered relevant. In our work, these document groups are represented as external user knowledge. As a way of obtaining such user knowledge, we use the relevance-feedback technique. Most users find it easier to answer ‘yes/no’ questions than to describe what they want. The clustering algorithm obtains feedback information from users as to whether retrieved documents are relevant or non-relevant to a submitted query, although it is not easy to obtain the full picture of a user’s intent.

Relevance feedback is the reformulation of a query in response to past assessments of relevance by the user. A query is expanded and reformulated by using additional terms to describe documents that the user finds relevant, while subtracting terms that are not. For query modification, we use the Ide Dec-Hi approach [?], which uses only the top non-relevant document for feedback, instead of all the non-relevant documents retrieved. With relevance feedback from the user, clustering performance can be further improved, although the browsing process required for deciding relevance can be quite time-consuming.

### 4 THE CLUSTERING METHOD

Our proposed clustering method basically incorporates relevance feedback into a traditional complete-linkage clustering algorithm in order to find more coherent clusters. The proposed method finds clusters in a given document collection using two steps. First, the document set is partitioned into relatively fine-grained clusters, whose diameter is less than a given threshold value. Next, the fine-grained clusters are repeatedly combined into genuine clusters by assigning them to those nearest to those of documents defined by the user as relevant. The overall algorithm proceeds in three phases: a pre-clustering phase, a supervising phase, and a re-clustering phase. The procedure for the algorithm is given in Figure 1.

#### 4.1 Pre-clustering phase: generation of fine-grained clusters

In the pre-clustering phase, the system clusters a given document collection at the fine-grained level. We call a cluster generated in this phase a ‘pre-cluster’. As in agglomerative hierarchical clustering, this algorithm starts with each data point as a separate cluster. Each step of the algorithm involves merging the two clusters whose merger produces the smallest increase in diameter, and after each merger, the total number of clusters decreases by one. These steps can be performed repeatedly until the diameter of the merged clusters approaches a given threshold. The number of resulting clusters is therefore determined by the cluster diameter threshold value.

The diameters of the clusters generated in this phase can be so adjusted to maximize the relevance of the documents within the fine-grained clusters. In this work, the value is set experimentally to a small non-zero value.

#### 4.2 Supervising phase: relevance-feedback interview

In the second phase, the user introduces one or more groups of relevant (or irrelevant) document examples to the clustering system, depending on his/her judgment of selected documents from a given corpus. We call this each document
group ‘document bundle’ or simply ‘bundle’. Here, we specify two types of document bundles: positive and negative ones. Documents within positive bundles (i.e., documents judged jointly relevant by users) are placed in the same cluster while documents within negative bundles must be located in the different clusters. In Figure 1, examples of positive bundles are \( \{d_1, d_3, d_5\} \) and \( \{d_2, d_6, d_11, d_13\} \), and an example of negative bundle is \( \{d_1, d_3\} \). Thus, even though the documents \( d_1 \) and \( d_{15} \) are very similar (i.e. the distance between both documents is short), these documents should not belong to the same cluster. Like this, the system is given a set of positive and negative bundles, which are denoted by \( D^+ \) and \( D^- \), respectively. Formally speaking, \( D^+ \) (or \( D^- \)) is a subset of power set of document set \( D \). During the relevance-feedback interview, there can be redundant documents within document bundles. However, any conflict among document bundles must not occur; in other words, any pair of documents within a positive bundle must not exist in any negative bundle, and vice versa.

In this work, the document bundles are developed by a relevance feedback program from interviews with the user. The interview program starts with some ‘training documents’. As seen in Figure 2, the training document set \( D' \) includes all documents within clusters of less than \( \eta \) documents. The program selects a document at random from this training set to submit a query. The system displays the query results for that query (document) to the user one at a time, to be either selected or ignored during cluster formation. Here, we may need some option that allows a user to better judge the relevance of a document to a concept.

We have observed that clusters created by subjects do not always share common words, because subjects cluster contents by meaning, not by keyword representation. Thus, in order that positive bundles can be described by more diverse vocabularies, it is preferable that the documents within each positive bundle has as small a number of shared words as possible. This helps to overcome the word mismatch problem and enhances the quality of clustering. To accomplish this, we use the fuzzy information retrieval method that was proposed in [2]. In this approach, documents relevant to a given query are retrieved by considering the degree of correlation between their index terms. The method retrieves those documents that have the most terms strongly correlated with the terms within the query.

In the fuzzy retrieval model, a fuzzy set \( F_x \) associated with each index term \( w_x \) is defined with Jaccard’s coefficient-style formula (3), which denotes a normalized correlation factor \( c_{x,y} \) between two terms \( w_x \) and \( w_y \). In this fuzzy set, a document \( d_j \) has a degree of membership \( \mu_{F_x}(d_j) \) computed by formula (4).

\[
c_{x,y} = \frac{d_f(x,y)}{d_f(x) + d_f(y) - d_f(x,y)} \tag{3}
\]

\[
\mu_{F_x}(d_j) = 1 - \prod_{w_x \in d_j} (1 - c_{x,y}) \tag{4}
\]

where \( d_f(x,y) \) is the number of documents that contain \( w_x(w_y) \), and \( d_f(x,y) \) is the number of documents that contain both terms. Since the fuzzy sets associated to each term are all fuzzy, a document \( d_i \) might belong to set \( F_x \), even if document \( d_i \) does not contain the term \( w_x \). The query algorithm \( \text{RetrieveFuzzyDocuments} \) for relevance feedback from training documents is given in Figure 3. At every feedback, the algorithm computes the membership \( \mu_{F_x}(d_i) \) of each document \( d_i \) of the training set in the fuzzy answer set \( F_q \) associated with the query document \( d_q \).

\[
\mu_{F_q}(d_i) = \prod_{w_x \in d_i} \mu_{F_x}(d_i)
\]

where

\[
\begin{align*}
\mu_{F_q}(d_i)' &= \mu_{F_q}(d_i) & \text{if } w_x \neq 0 \\
\mu_{F_q}(d_i)' &= 1 - \mu_{F_q}(d_i) & \text{if } w_x = 0
\end{align*}
\]

**Algorithm Clustering (Corpus D, Diameter \( \delta \))**

\[
\begin{align*}
D & \text{ is a given document corpus.} \nonumber \\
D^+ & \text{ (or } D^- \text{) is a subset of the positive (or negative) document bundles.} \nonumber \\
\delta & \text{ is the threshold value for the diameter of a cluster.} \nonumber \\
\{c\} & \text{ is a set of fine-level pre-clusters and its component } c_i \text{ is the } i\text{th pre-cluster.} \nonumber \\
\{C\} & \text{ is a set of final clusters and its component } C_i \text{ is the } i\text{th final cluster.} \nonumber \\
\{c\} & \text{ is the number of documents within the cluster } c_i. \nonumber \\
\{C\} & \text{ is a set of documents with the cluster } C_i. \nonumber \\
L_k & \text{ is the } k\text{th local prototype.} \nonumber \\
L_k & \text{ is a set of local prototypes belonging to the cluster } k. \nonumber \\
\begin{array}{l}
\text{begin} \nonumber \\
\text{1. Pre-clustering phase} \nonumber \\
\text{repeat} \nonumber \\
\{\text{Generate fine-level pre-clusters with diameter less than } \delta\} \nonumber \\
\{c_i, c_j\} = \text{clusters whose merger results in the small-}\nonumber \\
\text{est diameter } \delta \nonumber \\
c_{\text{new}} = \{c_i, c_j\} \nonumber \\
c = c \cup c_{\text{new}} - \{c_i, c_j\} \nonumber \\
\text{until } \text{diameter of } c_{\text{new}} < \delta \nonumber \\
\text{for each pre-cluster } c_i \in c \text{ do} \nonumber \\
\{\text{Let } D' \text{ be a set of training documents for relevance feedback.}\} \nonumber \\
\{\eta \text{ is the minimum number of documents required for}\} \nonumber \\
\text{a pre-cluster.} \nonumber \\
D' = D - \{c_i\} \text{ such that } |c_i| < \eta \nonumber \\
\text{for randomly selected document } d_i \in D' \text{ do} \nonumber \\
\{\text{Let } D^+_i(\{d_i\}) \text{ and } D^-_i(\{d_i\}) \text{ be the positive (or negative)}\} \nonumber \\
\text{document bundles that are retrieved from the document} \nonumber \\
\text{query document } d_i. \nonumber \\
\{D^+_i(\{d_i\}), D^-_i(\{d_i\})\} = \text{RetrieveFuzzyDocuments}(d_i, D') \nonumber \\
\{C'\} = D' \cup \{d_i\} \nonumber \\
D' = D' - D^-_i(\{d_i\}) \nonumber \\
D^-_i = D^-_i \cup D^-_i(\{d_i\}) \nonumber \\
\text{end} \nonumber \\
\text{2. Supervising phase} \nonumber \\
\text{for each pre-cluster } c_i \in c \text{ do} \nonumber \\
\text{let } D' \text{ be a set of training documents for relevance feedback.} \nonumber \\
\{\text{Let } \eta \text{ be the minimum number of documents required for a pre-cluster.}\} \nonumber \\
D' = D - \{c_i\} \text{ such that } |c_i| < \eta \nonumber \\
\text{for randomly selected document } d_i \in D' \text{ do} \nonumber \\
\{\text{Let } D^+_i(\{d_i\}) \text{ and } D^-_i(\{d_i\}) \text{ be the positive (or negative)}\} \nonumber \\
\text{document bundles that are retrieved from the} \nonumber \\
\text{query document } d_i. \nonumber \\
\{D^+_i(\{d_i\}), D^-_i(\{d_i\})\} = \text{RetrieveFuzzyDocuments}(d_i, D') \nonumber \\
\{C'\} = D' \cup \{d_i\} \nonumber \\
D' = D' - D^-_i(\{d_i\}) \nonumber \\
D^-_i = D^-_i \cup D^-_i(\{d_i\}) \nonumber \\
\text{end} \nonumber \\
\text{3. Re-clustering phase} \nonumber \\
\{\text{Pre-cluster } c_i \text{ is assigned to the nearest cluster } C_k \} \nonumber \\
\text{for each pre-cluster } c_i \in c \text{ do} \nonumber \\
\text{begin} \nonumber \\
N N(c_i) = \arg\min \{d_j - c_i\} \nonumber \\
C_k = C_k \cup c_i \text{ such that } N N(c_i) \in C_k \nonumber \\
\text{Compute the local prototype of the updated cluster} \nonumber \\
\text{if } \{L_k, N N(c_i)\} \in L_k \text{ then update } L_k \nonumber \\
\text{else } L_k = L_k \cup \{L_k, N N(c_i)\} \nonumber \\
\text{end} \nonumber \\
\{\text{Assign the rest of the training documents to the nearest (local)}\} \nonumber \\
\text{prototype } \} \nonumber \\
\text{for each document } d_i \in D' \text{ do} \nonumber \\
\text{begin} \nonumber \\
\{C\} = \{C\} \cup \{d_i\} \text{ such that all pairs of the documents} \nonumber \\
\text{within } \{C\} \text{ are not included in any document} \nonumber \\
\text{bundle in } D^-_i, \text{ and a local prototype of } C_j \nonumber \\
\text{is the nearest to } d_i \nonumber \\
\text{end} \nonumber \\
\text{end} \nonumber \
\end{array}
\]
In the RetrieveFuzzyDocuments algorithm that performs the relevance-feedback interview, one document \( d_i \) is randomly selected from documents in the training set \( D' \). Then, regarding the document as a query, the relevance feedback interview with the user proceeds. In this work, the frequency of feedback is set to three times, since several previous related works, such as [2] and [3], have stated that most relevant documents can be retrieved within this frequency. This relevance feedback procedure produces a set of seeds for each genuine cluster, which becomes the basis of its multiple prototype. Let \( D_i^+(d_1), D_i^+(d_2), \ldots \) be positive document bundles relevant to the randomly-selected query documents \( d_1, d_2, \ldots \), respectively. Each \( D_i^+(d_i) \) becomes a basis of requested clusters. Therefore, the final number of resulting clusters is determined by the number of times that the user performs the feedback interview.

4.3 Re-clustering phase: assigning a pre-cluster to its nearest positive document bundle

The final phase of the clustering process involves the formation of genuine clusters in the entire document collection. The pre-clusters \( (c_i) \) created in the first phase are assigned to the positive bundle in which their nearest document \( (NN(c_i)) \) is found. At every assignment of pre-clusters, the set of local prototypes \( (L_k) \) of clusters is incrementally updated. As a result, those pre-clusters get around the document seeds within the positive bundle and larger clusters are then generated. In the context of text classification, training data is formed through relevance feedback and the pre-clusters are checked as a test set. Finally, residual documents that have not been retrieved or ignored during the relevance feedback process are assigned to the cluster that has their nearest local prototype. At this time, documents within the negative bundles are examined whether they are located in the same clusters. If such documents are found, they should be placed in the different clusters. To achieve this, the document newly assigned to the cluster is re-assigned to another cluster that has its second nearest local prototype.

5. EMPIRICAL RESULTS

In this section, we study the performance of the proposed method and demonstrate its effectiveness for clustering compared to that of the complete-linkage clustering method.

5.1 Experimental Setup

To evaluate our method, we used a controlled subset of the Reuters-21578 collection, which has been accepted as a clean corpus and is frequently used to test different algorithms for text categorization. First, we selected the documents belonging to the 7 most frequent topics, such as earn, aqc, money-fx, grain, crude, trade, interest, and ship. From among these selected documents, we chose those that had a single topic, thus avoiding the ambiguity of documents with multiple topics. This generated 5 test sets are shown in Table 1. For the performance evaluation to be reliable, we generated test sets with different numbers of topics. Each test set includes documents for at least three topics. The documents for each set are randomly selected from the Reuters-21578 corpus and cannot overlap with the documents of other data sets.

The vector space model suffers from the high dimensionality of the large volume of terms present. Usually, a classification or clustering method requires that only the more differentiating words be used for efficient execution. Therefore, we apply both a stop word elimination and a Zipf’s law-based feature selection to the controlled data sets. The individual terms in the corpus were extracted, and the stop words were removed using a stop word list [?] consisting of more than 500 common English words.

Furthermore, for feature reduction (or selection) we simply use feature selection based on Zipf’s law. According to Zipf distribution, terms that have the lowest frequency of occurrence in a corpus are not helpful to document clustering or classification [?]. Thus we eliminate any term that appears fewer than 5 times in a given test set. The number of words reported in Table 1 reflects the number of words in each data set after such feature selection.

5.2 Performance Metrics

We discuss the following results with respect to the F-measure metric. A subset of documents corresponding to a selected topic should appear as a cluster in the organization. Therefore, we compare how closely each cluster generated by the clustering algorithm matches the set of categories previously assigned to the documents by human judges. We use the standard measurements of information retrieval performance: precision, recall and F-measure for each cluster, with respect to the topic in question. These are denoted as \( p \), \( r \) and \( F \), respectively. Precision in document clustering indicates the consistency of the documents within a cluster. Recall provides an indication of the number of document be-

---

Figure 2: Clustering Algorithm

Figure 3: Fuzzy Retrieval Algorithm
For any topic \( t \), and are proportionally related to clustering effectiveness. We use the F1-measure, which gives equal weight to both recall and precision. These three measures vary from 0 to 1, and are proportionally related to clustering effectiveness. For any topic \( t \) and cluster \( i \):

\[
F_{i,t} = \frac{2 \cdot p_{i,t} \cdot r_{i,t}}{p_{i,t} + r_{i,t}}
\]

\[
F_i = \max_{t} F_{i,t}
\]

Using the F1-measure for each topic, we compute the overall F1-measure for the collection. To do so, we decide which cluster has the highest F1-Measure, and that F1-Measure, \( F_i \), becomes the system’s score for topic \( t \). The overall F-Measure \( F_{total} \) is computed as the weighted average of the F-Measures for each topic \( t \) as follows:

\[
F_{total} = \frac{\sum_{i \in T} |t| \cdot F_i}{\sum_{i \in T} |t|}
\]

where \( T \) is the set of topics, \( |t| \) is the number of documents judged to be on topic \( t \).

### 5.3 Evaluation of Clustering Results

We used the complete-linkage clustering method as a baseline for performance measurement. In this section, we focus on evaluating the contribution of the user’s external knowledge to the clustering quality, and the extent to which the size of a pre-cluster determines quality. Our experimental results show that incorporating the user’s external knowledge into the clustering process yields a significant improvement in performance.

#### 5.3.1 Effect of the degree of supervision

To evaluate the contribution of relevance feedback, we first need to define the degree of supervision, which presents the user’s effort in guiding the clustering process. In this work, this is defined to be

\[
s = \frac{|D^r|}{|D|}
\]

where \( |D^r| \) is the number of training documents excluded from a set of pre-clusters and \( |D| \) is the total number of documents retrieved by relevance-feedback interview. The degree of supervision is 1 when the user classifies all the training documents through relevance feedback.

As might be expected, the clustering algorithm yields better-quality clusters when external knowledge is incorporated into the clustering process. Figure 4 shows that a small number of defined relevant groups, covering about 30-40% of the documents, is enough to improve the performance of the clustering system. As the degree of supervision approaches that point, clustering quality increases rapidly. This result indicates that even a little external knowledge provides the clustering process with valuable leads to topical structures in the test sets. For the controlled test sets with different numbers of topics, our method provided significantly better results than the complete-linkage method, with an average increase of 30% in the F1-measure. Figure 5 illustrates the topic distribution of the documents in each cluster for the best performance with a sufficiently large F1-measure value. Documents within a topic are much more fragmented among the resulting clusters using complete-linkage clustering while the proposed method generates clusters that contain a set of relatively coherent documents.

#### 5.3.2 Effect of the size of a pre-cluster

In this section, we describe the effect of the size of pre-clusters on clustering. When the cluster diameter threshold value is below a certain value, the proposed method provides good clusters, since pre-clusters with a smaller diame-

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of topics</th>
<th>Dataset size</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>T3</td>
<td>3</td>
<td>369 × 2,666</td>
<td>earn(123) acq(153) trade(93)</td>
</tr>
<tr>
<td>T4</td>
<td>4</td>
<td>381 × 2,578</td>
<td>earn(215) grain(20) ship(35) trade(111)</td>
</tr>
<tr>
<td>T5</td>
<td>5</td>
<td>379 × 2,834</td>
<td>earn(126) acq(133) interest(43) grain(20) crude(57)</td>
</tr>
<tr>
<td>T6</td>
<td>6</td>
<td>373 × 2,670</td>
<td>earn(95) acq(87) crude(40) trade(47) interest(43) money-fx(33)</td>
</tr>
<tr>
<td>T7</td>
<td>7</td>
<td>384 × 2,267</td>
<td>earn(95) acq(88) money-fx(40) crude(43) trade(40) interest(43) ship(35)</td>
</tr>
</tbody>
</table>

Figure 4: The effects of supervision degree on clustering quality

<table>
<thead>
<tr>
<th>Supervision Degree</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.4</td>
<td>0.5</td>
<td>0.6</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Table 1: Setup of test document sets
Figure 5: Distribution of topics within clusters

Figure 6: The effects of diameter changes on clustering quality: The ‘PROPOSED’ means the proposed supervised clustering method and the ‘AHC’ denotes the complete-linkage AHC clustering method

6. CONCLUSIONS AND FUTURE WORK

We have presented a method for clustering documents that is based on partially supervised information organization. This method enables the partitioning of a corpus based on the user’s perception of similarity between documents, rather than relying only on a given similarity function. Our experiments indicate that the proposed clustering method holds significant promise for isolating topically coherent document groups, given a little human effort through relevance-feedback interview.

We are continuing to investigate techniques for optimizing the diameter of pre-clusters, based on our expectation that coherent clusters can have a larger threshold value when the association between terms is strong. The threshold value could then be determined according to the degree of correlation between terms in the documents within the clusters. We are also trying to refine the process of knowledge acquisition during the feedback interview, so that users can easily compile the document bundles. Ultimately, we plan to build a semi-automatic document indexing system for user-driven organization of very large amounts of information. We anticipate that this system will not only allow users to easily develop new organizational schemes, but will also help them to maintain extensible hierarchies of categories. The proposed clustering algorithm can be effectively used as a primitive function in isolating closely related documents in a hierarchical structure.

7. REFERENCES


