

# Topic-driven Clustering for Document Datasets \*

Ying Zhao <sup>†</sup>

George Karypis <sup>‡</sup>

## Abstract

In this paper, we define the problem of *topic-driven clustering*, which organizes a document collection according to a given set of topics. We propose three topic-driven schemes that consider the similarity between documents and topics and the relationship among documents themselves simultaneously. We present a comprehensive experimental evaluation of the proposed topic-driven schemes on five datasets. Our experimental results show that the proposed topic-driven schemes are efficient and effective with topic prototypes of different levels of specificity.

## 1 Introduction

Fast and high-quality document clustering algorithms play an important role in providing intuitive navigation and browsing mechanisms by organizing large amounts of information into a small number of meaningful clusters. As unsupervised learning methods, clustering algorithms do not require any prior knowledge of the datasets in general. However, when such prior knowledge is available, clustering algorithms should also be able to benefit from it to produce more desired clustering solutions. In particular, we focus on the type of the prior knowledge that reflects the cognition of the natural clusters by domain experts. For example, in many knowledge management applications, even though the complete taxonomy of the document collection is not available, often times domain experts can describe the major topics (clusters) that the collection covers. Moreover, they would like the clustering algorithms produce

the clustering solutions that are consistent with their cognition models. Hence it is important to be able to organize a document collection according to a given set of topics (either from domain experts, or as a requirement satisfying users' needs). We refer to this problem as *topic-driven clustering*.

For example, in a typical environment of law firms, a large amount of letters, memoranda, e-mail messages, and contracts are generated on a daily basis. Thus, organizing legal documents into meaningful clusters to leverage browsing and searching is very important. Even though a complete law firm taxonomy may not be available, law librarians in a law firm can provide some information on the major topics of the document collection based on their knowledge on the practice areas of the law firm, related law categories, and custom base. For example, a law firm may focus on the areas of banking, bankruptcy, insurance, and debt. This information is not only helpful for organizing the documents, but also serves as a requirement of desired clustering solutions. That is, the resultant clusters should correspond to the identified topics (for example, the four practice areas). Also note that acquiring prior knowledge of labeled documents associated with each topic, even a small amount, can be very time-consuming and costly.

The traditional classification algorithms cannot solve the topic-driven clustering problem because of the insufficient information about each class (topic). Since sometimes the available descriptions may only contain a few words, in order to produce good organization, the information of unlabeled documents must be taken into account to leverage classification technology. Semi-supervised classification [13, 3] and active learning [6, 23] are two of such approaches. However, these approaches either need sufficient labeled data to start with, or need to have access to a nontrivial amount of labeled data during the process.

The current approaches of semi-supervised clustering [25, 1, 26, 11], which can use class labels or pairwise constraints on some documents during the clustering process, fail to satisfy the requirements of the topic-driven clustering problem as well, mainly because the prior knowledge of the topic-driven clustering problem is not in the format of labeled objects, but of the de-

\*This work was supported in part by NSF CCR-9972519, EIA-9986042, ACI-9982274, ACI-0133464, and ACI-0312828; the Digital Technology Center at the University of Minnesota; and by the Army High Performance Computing Research Center (AHPCRC) under the auspices of the Department of the Army, Army Research Laboratory (ARL) under Cooperative Agreement number DAAD19-01-2-0014. The content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute.

<sup>†</sup>Dept. of Computer Science and Engineering, University of Minnesota, MN

<sup>‡</sup>Dept. of Computer Science and Engineering, University of Minnesota, MN

scriptions of possible topics.

We propose the topic-driven clustering schemes to solve this problem by defining two properties that a good clustering solution must have. First, the documents clustered to a certain topic should contain the content of the topic (*i.e.*, the documents are similar or relevant to the topic). Second, the documents within one cluster should be more similar to each other than the documents from two different clusters.

The contribution of this paper is two-fold. First, to our knowledge, we introduce this novel problem of topic-driven clustering. Second, we propose effective and efficient topic-driven clustering methods that emphasize the relationship between documents and topics and relationship among documents themselves simultaneously. In addition, we present a comprehensive experimental evaluation using various datasets and our experimental results show that the proposed topic-driven clustering schemes are effective with topic prototypes of different levels of specificity.

The rest of this paper is organized as follows. Section 2 discusses some related research. Section 3 provides some information on how documents are represented and how the similarity or distance between documents is computed. Section 4 describes the criterion functions that are the focus of this paper and describes the algorithm that optimizes the various topic-driven criterion functions and the clustering algorithm itself. Section 5 provides the detailed experimental evaluation of the various topic-driven criterion functions. Finally, Section 6 provides some concluding remarks.

## 2 Related Research

Active learning [6, 23] acknowledges the fact that acquiring labeled data is very time-consuming and costly, and tries to minimize the number of labeled data required to build a successful classifier. The active learning approaches start with a very small number of labeled data and request unlabeled objects to be labeled based on whether the unlabeled objects are “more informative” point. The active learning approaches utilize the information provided by unlabeled data. However, they still need to have access to sufficient labeled data.

Incorporating prior knowledge into the clustering process has drawn people’s attention recently. The focus of this research has been on semi-supervised clustering, which assumes the prior knowledge (background knowledge) is given by a limited set of labeled data, from which the knowledge of two objects should belong to the same cluster (must-link) or should not belong to the same cluster (cannot-link) can be derived. Previous semi-supervised approaches fall into three categories: constraint-based, metric-based and the combined ap-

proaches. Constraint-based approaches explicitly modify the objective function or make certain constraints during the clustering process[25]. Whereas, metric-based approaches parametrize distance metric and learn the metric parameters in a manner, so that the distance between objects connected by must-links is smaller and the distance between objects connected by cannot-links is larger in general [1, 26]. Finally the combined approaches integrate both of these techniques in the clustering process [2]. Another recent approach of incorporating prior knowledge tackles the problem differently [11]. They defined the non-redundant data clustering as a problem of discovering alternative clustering solutions given a known clustering solution, where the prior knowledge is an entire clustering solution.

## 3 Preliminaries

Through-out this paper we will use the symbols  $n$ ,  $m$ , and  $k$  to denote the number of documents, the number of terms, and the number of clusters, respectively. We will use the symbol  $S$  to denote the set of  $N$  documents that we want to cluster,  $S_1, S_2, \dots, S_k$  to denote each one of the  $k$  clusters,  $n_1, n_2, \dots, n_k$  to denote the sizes of the corresponding clusters, and  $T_1, T_2, \dots, T_k$  to denote the topic prototype vectors given as prior knowledge.

The various clustering algorithms that are described in this paper use the vector-space model [20] to represent each document. In this model, each document  $d$  is considered to be a vector in the term-space. In particular, we employed the *tf-idf* term weighting model, in which each document can be represented as

$$(tf_1 \log(n/df_1), tf_2 \log(n/df_2), \dots, tf_m \log(n/df_m)).$$

where  $tf_i$  is the frequency of the  $i$ th term in the document and  $df_i$  is the number of documents that contain the  $i$ th term. To account for documents of different lengths, the length of each document vector is normalized so that it is of unit length ( $\|d_{tfidf}\| = 1$ ), that is each document is a vector on the unit hypersphere. In the rest of the paper, we will assume that the vector representation for each document and for each topic has been weighted using *tf-idf* and it has been normalized so that it is of unit length. Given a set  $A$  of documents and their corresponding vector representations, we define the **composite** vector  $D_A$  to be  $D_A = \sum_{d \in A} d$ , and the **centroid** vector  $C_A$  to be  $C_A = D_A/|A|$ . We also define the composite vector of the entire dataset to be  $D = \sum_{i=1}^N d_i$ , and the composite vector of the entire topics to be  $T = \sum_{i=1}^k T_i$ .

In the vector-space model, the cosine similarity is the most commonly used method to compute the similarity between two documents  $d_i$  and  $d_j$ , which is defined to be  $\cos(d_i, d_j) = d_i^t d_j / (\|d_i\| \|d_j\|)$ . The cosine

formula can be simplified to  $\cos(d_i, d_j) = d_i^t d_j$ , when the document vectors are of unit length. This measure becomes one if the documents are identical, and zero if there is nothing in common between them (*i.e.*, the vectors are orthogonal to each other).

## 4 Topic-driven Clustering Algorithms

At a high-level the problem of topic-driven clustering is defined as follows. Given a set  $S$  of  $n$  documents and a set  $T$  of  $k$  topics, we would like to partition the documents into  $k$  subsets  $S_1, S_2, \dots, S_k$ , each corresponding to one of the topics, such that (i) the documents assigned to each subset are more similar to each other than the documents assigned to different subsets, and (ii) the documents of each subset are more similar to its corresponding topic than the rest of the topics.

Even though there are a number of different ways that can be used to convert the above high-level problem definition into a precise clustering algorithm, in this paper, we will limit our focus to the class of algorithms that use a global criterion function  $\mathcal{C}$  to capture the properties and quality of the desired clustering solution and formulate the clustering problem as that of an optimization problem that tries to compute a  $k$ -way clustering solution such that the value of  $\mathcal{C}$  is optimized [10].

In the rest of this section we first present a number of different criterion functions that can represent the requirements of the topic-driven clustering problem, followed by a description of the algorithms that were used to perform their optimization.

### 4.1 Criterion Functions

Since the requirements of the topic-driven clustering contain two components, we first look at how to model them separately. The first component emphasizes the relationship between documents and tries to guide the clustering process to produce clustering solutions in which documents from the same cluster are more similar to each other than the documents assigned to different clusters. This component does not consider the topics and we will refer to the criterion functions that fall into this category as *unsupervised criterion functions*. On the other hand, the second component emphasizes whether the documents in each cluster are relevant to the topic associated with the cluster. We will refer to the criterion functions in this category as *supervised criterion functions*. In the rest of this section we will first discuss several criterion functions from each category and then propose two schemes to combine them together. At the end, we propose the third scheme, which is a hybrid approach that incorporates the two aspects into a single criterion function.

#### 4.1.1 Unsupervised Criterion Functions

People have proposed a great number of criterion functions in this category over the past few years [7, 15, 8, 9]. Recently, we [31, 30] studied eight different partitioned clustering criterion functions in the context of document clustering, which optimize various aspects of intra-cluster similarity, inter-cluster dissimilarity, and their combinations. Our experiments revealed that different criterion functions lead to substantially different results, whereas our analysis showed that their performance depends on the degree to which they can correctly operate when the dataset contains clusters of different densities (*i.e.*, they contain documents whose pairwise similarities are different) and the degree to which they can produce balanced clusters.

In this study, we focus on two internal criterion functions ( $\mathcal{I}_1$  and  $\mathcal{I}_2$ ) and one external criterion function ( $\mathcal{E}_1$ ) [29, 31]. This subset represents some of the most widely-used criterion functions for document clustering, and includes some of the best- and worst-performing schemes.

The  $\mathcal{I}_1$  criterion function (Equation 4.1) maximizes the sum of the average pairwise similarities (as measured by the cosine function) between the documents assigned to each cluster weighted according to the size of each cluster and has been used successfully for clustering document datasets [19].

$$\mathcal{I}_1 = \sum_{r=1}^k n_r \left( \frac{1}{n_r^2} \sum_{d_i, d_j \in S_r} \cos(d_i, d_j) \right)$$

$$(4.1) \quad \mathcal{I}_1 = \sum_{r=1}^k \frac{\|D_r\|^2}{n_r}.$$

The  $\mathcal{I}_2$  criterion function (Equation 4.2) is used by the popular vector-space variant of the  $K$ -means algorithm [7, 15, 8, 22]. In this algorithm each cluster is represented by its centroid vector and the goal is to find the solution that maximizes the similarity between each document and the centroid of the cluster that is assigned to.

$$(4.2) \quad \mathcal{I}_2 = \sum_{r=1}^k \sum_{d_i \in S_r} \cos(d_i, C_r) = \sum_{r=1}^k \|D_r\|.$$

The  $\mathcal{E}_1$  criterion function (Equation 4.3) computes the clustering by finding a solution that separates the documents of each cluster from the entire collection. Specifically, it tries to minimize the cosine between the centroid vector of each cluster and the centroid vector of the entire collection. The contribution of each cluster is weighted proportionally to its size so that larger clusters

will be weighted higher in the overall clustering solution.  $\mathcal{E}_1$  was motivated by multiple discriminant analysis and is similar to minimizing the trace of the between-cluster scatter matrix [10].

$$(4.3) \quad \mathcal{E}_1 = \sum_{r=1}^k n_r \cos(C_r, C) \Leftrightarrow \sum_{r=1}^k n_r \frac{D_r^t D}{\|D_r\|}.$$

#### 4.1.2 Supervised Criterion Functions

For the topic-driven clustering problem, we assume that the description of each cluster is available as prior knowledge and can be represented as a vector. Given these cluster prototype vectors, the similarity between each document and its topic can be defined as the cosine similarity between the vector of the document  $d$  and the prototype vector of the topic  $T_r$ . Hence, we can define internal and external supervised criterion functions similar to the unsupervised criterion functions.

The internal supervised criterion function, denoted by  $\mathcal{S}_{\mathcal{I}}$ , tries to maximize the similarity between the documents in a cluster to the topic associated with the cluster. The formal definition can be written as

$$(4.4) \quad \mathcal{S}_{\mathcal{I}} = \sum_{r=1}^k \sum_{d_i \in \mathcal{S}_r} \cos(d_i, T_r) = \sum_{r=1}^k D_r^t T_r.$$

The external supervised criterion function, denoted by  $\mathcal{S}_{\mathcal{E}}$ , tries to minimize the similarity between each document to the topics that are not associated with its cluster. Let  $\bar{T}_r = T - T_r$ , then the external supervised criterion function ( $\mathcal{S}_{\mathcal{E}}$ ) can be written as

$$(4.5) \quad \mathcal{S}_{\mathcal{E}} = \sum_{r=1}^k \sum_{d_i \in \mathcal{S}_r} \cos(d_i, \bar{T}_r) = \sum_{r=1}^k D_r^t \bar{T}_r / \|\bar{T}_r\|.$$

Note that since maximizing  $\sum_{r=1}^k D_r^t T_r$  is the same as minimizing  $\sum_{r=1}^k D_r^t \bar{T}_r$ , the difference between  $\mathcal{S}_{\mathcal{I}}$  and  $\mathcal{S}_{\mathcal{E}}$  is that in  $\mathcal{S}_{\mathcal{E}}$  the dissimilarities between documents to the other topics is scaled by the norm of the composite of the other topics.

#### 4.1.3 Combined Criterion Functions

Combining unsupervised and supervised criterion functions can be treated as a multi-objective optimization problem, which has been studied in many different domains [14, 28, 21]. One of the real difficulties in this problem is that no single optimal solution exists. Instead, an optimal solution exists for each objective in

the solution space. The result is that the definition of a good solution becomes ambiguous. Thus, we need to develop a scheme that can disambiguate the definition of a good solution. A good scheme should allow fine-tuned control of the tradeoffs among the objectives and be able to handle objectives that correspond to quantities that are both of similar as well as of different types.

One straightforward means of disambiguating the definition of a good multi-objective solution is to assign the objectives different weights before combining them together, which we refer to as the *weighted* scheme.

Given two criterion functions  $X$  and  $Y$ , the weighted scheme can be written as

$$(4.6) \quad M_1(X, Y) = \alpha X + (1 - \alpha)Y,$$

where  $\alpha$  is the preference factor.

The weighted scheme allows a fine-tuned control of the tradeoffs among the objectives by varying the preference factor  $\alpha$ . However, this formulation cannot handle dissimilar criterion functions or the criterion functions that change in different scales, because a weighted sum of them can be meaningless.

Deriving topic-driven criterion functions based on the weighted scheme can be done easily for  $\mathcal{I}_2$  and  $\mathcal{E}_1$ , since both  $\mathcal{I}_2$  and  $\mathcal{E}_1$  have  $N$  terms in their formulas. However, for  $\mathcal{I}_1$ , to make it also contain  $N$  terms as in  $\mathcal{S}_{\mathcal{I}}$ , we need to multiply the  $\mathcal{I}_1$  function by  $N$  before combining it with  $\mathcal{S}_{\mathcal{I}}$ .

Motivated by the method of combining multiple objective functions in graph partitioning [21], we propose the second scheme, the *normalized* scheme. Our formulation is based on the intuitive notion of what constitutes a good multi-objective solution. Quite often, a natural way of evaluating the quality of a multi-objective solution is to look at how close it is to the optimal solution of each individual objective. Hence, before combining two criterion functions, we normalize them with the optimal values that can be achieved by optimizing the two criterion functions separately.

Given two criterion functions  $X$  and  $Y$ , let  $X^*$  and  $Y^*$  denote the criterion function values of the optimal solutions with respect to  $X$  and  $Y$ , respectively, then the normalized scheme of combining two criterion functions can be defined as

$$(4.7) \quad M_2(X, Y) = \alpha \frac{X}{X^*} + (1 - \alpha) \frac{Y}{Y^*},$$

where  $\alpha$  is the preference factor.

In essence, optimizing Equation 4.7 attempts to compute a  $k$ -way clustering such that the criterion function values with respect to each criterion are not far away from the optimal values. This scheme works with both similar and dissimilar objectives. This is

because it makes all quantities similar before combining them. Each criterion function value is divided by the optimal value of its corresponding criterion, and so, represents a certain fraction of the optimal value. Since all components now represent a fraction of the optimal value, they can be combined meaningfully.

Finally, the various topic-driven criterion functions derived by the two combined schemes are shown in Table 1, the clustering problem becomes that of maximizing  $M_1(\mathcal{I}_1)$ ,  $M_1(\mathcal{I}_2)$ ,  $M_2(\mathcal{I}_1)$ , and  $M_2(\mathcal{I}_2)$ , and minimizing  $M_1(\mathcal{E}_1)$  and  $M_2(\mathcal{E}_1)$  accordingly.

#### 4.1.4 Hybrid Criterion Functions

For  $\mathcal{I}_1$  and  $\mathcal{I}_2$ , we propose the third scheme that incorporates the two aspects into a single criterion function. The motivation behind this hybrid scheme is that we noticed the relationship between the unsupervised criterion function  $\mathcal{I}_2$  and the internal supervised criterion function  $\mathcal{S}_{\mathcal{T}}$ . The former maximizes the summation of the similarity of each document to its cluster centroid and the latter maximizes the summation of the similarity of each document to its cluster topic. If we could define a new center that represents both the cluster centroid and the topic, then we could maximize the summation of the similarity of each document to this new center and address the two requirements of the topic-driven clustering problem at the same time. To this end, we define the *topic-weighted* composite vector of the  $r$ th cluster  $C'_r$  as  $D'_r = \sum_{d \in \mathcal{S}_r} \cos(d, T_r) d_i$  and the weighted size  $n'_r = \sum_{d \in \mathcal{S}_r} \cos(d, T_r)$ . The *topic-weighted* centroid can be defined as

$$C'_r = \frac{\sum_{d \in \mathcal{S}_r} \cos(d, T_r) d_i}{n'_r},$$

which takes into account the similarity of each document to its cluster topic. Using the above definition, the hybrid  $\mathcal{I}_2$  criterion function, denoted by  $H(\mathcal{I}_2)$ , can be obtained by requiring the clustering solution to maximize the similarity between the documents assigned to a cluster and its topic-weighted centroid. This is formally defined as follows:

$$(4.8) \quad H(\mathcal{I}_2) = \sum_{r=1}^k \sum_{d_i \in \mathcal{S}_r} \cos(d_i, C'_r) = \sum_{r=1}^k D_r^t C'_r$$

Similarly, the  $\mathcal{I}_1$  can be rewritten as

$$\mathcal{I}_1 = \sum_{r=1}^k \sum_{d_i \in \mathcal{S}_r} \frac{d_i^t D_r}{n_r}.$$

If we use the topic-weighted composite and size of the  $r$  cluster to replace the composite and size, we can get

the hybrid  $\mathcal{I}_1$  criterion function, denoted by  $H(\mathcal{I}_1)$ , as follows:

$$(4.9) \quad H(\mathcal{I}_1) = \sum_{r=1}^k \sum_{d_i \in \mathcal{S}_r} \frac{d_i^t D'_r}{n'_r}.$$

Given the formulation of  $H(\mathcal{I}_1)$  and  $H(\mathcal{I}_2)$ , the clustering problem becomes that of finding the clustering solutions that maximize  $H(\mathcal{I}_1)$  and  $H(\mathcal{I}_2)$ , respectively.

## 4.2 Partitional Clustering Algorithm

The partitional method we used to optimize the various criterion functions is very similar and also similar to that used in [22, 29]. Our optimizer computes the clustering solution by first obtaining an initial  $k$ -way clustering and then applying an iterative refinement algorithm to further improve it. The algorithms that optimize unsupervised, supervised and topic-driven criterion functions differ in two ways: whether topic vectors are used as initial seeds; and whether topic vectors are allowed to move to another cluster.

### 4.2.1 Initialization

We employed two different ways of producing the initial clustering. For optimizing unsupervised criterion functions, during initial clustering,  $k$  documents are randomly selected to form the *seeds* of the clusters and each document is assigned to the cluster corresponding to its most similar seed.

For the various supervised and topic-driven criterion functions, the  $k$  topic vectors are used as the initial seeds for the  $k$  clusters and each document is assigned to the cluster corresponding to its most similar seed.

### 4.2.2 Optimization Methods

The refinement strategy that we used consists of a number of iterations. During each iteration, the documents are visited in a random order. For each document,  $d_i$ , we compute the change in the value of the criterion function obtained by moving  $d_i$  to one of the other  $k - 1$  clusters. If there exist some moves that lead to an improvement in the overall value of the criterion function, then  $d_i$  is moved to the cluster that leads to the highest improvement. If no such cluster exists,  $d_i$  remains in the cluster that it already belongs to. The refinement phase ends, as soon as we perform an iteration in which no documents moved between clusters. Note that unlike the traditional refinement approach used by  $K$ -means type of algorithms, the above algorithm moves a document as soon as it is determined that it will lead to an improvement in the value of the criterion function. This type of refinement algorithms are often called *incremental* [10]. Since each move directly optimizes the particular crite-

Table 1: Clustering Criterion Functions.

Weighted Scheme	
$M_1(\mathcal{I}_1) = M_1(N\mathcal{I}_1, \mathcal{S}_{\mathcal{I}}) = \alpha N \sum_{r=1}^k \frac{\ D_r\ ^2}{n_r} + (1 - \alpha) \sum_{r=1}^k D_r^t T_r$	
$M_1(\mathcal{I}_2) = M_1(\mathcal{I}_2, \mathcal{S}_{\mathcal{I}}) = \alpha \sum_{r=1}^k \ D_r\  + (1 - \alpha) \sum_{r=1}^k D_r^t T_r$	
$M_1(\mathcal{E}_1) = M_1(\mathcal{E}_1, \mathcal{S}_{\mathcal{E}}) = \alpha \sum_{r=1}^k \frac{D_r^t D}{\ D_r\  \ D\ } + (1 - \alpha) \sum_{r=1}^k D_r^t \bar{T}_r$	
Normalized Scheme	
$M_2(\mathcal{I}_1) = M_2(N\mathcal{I}_1, \mathcal{S}_{\mathcal{I}}) = \alpha \frac{\sum_{r=1}^k \frac{\ D_r\ ^2}{n_r}}{\mathcal{I}_1^*} + (1 - \alpha) \frac{\sum_{r=1}^k D_r^t T_r}{\mathcal{S}_{\mathcal{I}}^*}$	
$M_2(\mathcal{I}_2) = M_2(\mathcal{I}_2, \mathcal{S}_{\mathcal{I}}) = \alpha \frac{\sum_{r=1}^k \ D_r\ }{\mathcal{I}_2^*} + (1 - \alpha) \frac{\sum_{r=1}^k D_r^t T_r}{\mathcal{S}_{\mathcal{I}}^*}$	
$M_2(\mathcal{E}_1) = M_2(\mathcal{E}_1, \mathcal{S}_{\mathcal{E}}) = \alpha \frac{\sum_{r=1}^k \frac{D_r^t D}{\ D_r\  \ D\ }}{\mathcal{E}_1^*} + (1 - \alpha) \frac{\sum_{r=1}^k D_r^t \bar{T}_r}{\mathcal{S}_{\mathcal{E}}^*}$	

rion function, this refinement strategy always converges to a local minima.

Note that for the various supervised and topic-driven criterion functions, it is important to keep the topic vector always associate with its own cluster. Hence we do not allow the topic vector move to other clusters, and the clustering problem becomes that of forming clusters around the topic vectors.

The optimization method for the normalized scheme is different from the others, because it requires the optimal criterion function values obtained by optimizing the two criterion functions separately before performing the optimization of the combined criterion functions. Hence, the optimization method for the normalized scheme contains three rounds of refinement. The first two rounds optimize the two individual criterion functions, and the third round starts from the same initial clustering and uses the optimal criterion function values achieved in the first two rounds as the normalization factors.

The greedy nature of the refinement algorithm does not guarantee that it will converge to a global optima, and the local optima solution it obtains depends on the particular set of seed documents that were selected during the initial clustering. To eliminate some of this sensitivity, the overall process is repeated a number of times. That is, we compute  $N$  different clustering solutions (*i.e.*, initial clustering followed by cluster refinement), and the one that achieves the best value for the particular criterion function is kept. In all of our experiments, we used  $N = 10$ . For the rest of this discussion when we refer to the clustering solution we will mean the solution that was obtained by selecting the best out of these  $N$  potentially different solutions.

### 4.2.3 Computational Complexity

One of the advantages of our partitional algorithm and that of other similar partitional algorithms, is that it has relatively low computational requirements. A  $k$ -way clustering of a set of documents can be computed in time linear on the number of documents and the number of clusters  $k$ , as in most cases the number of iterations required by the greedy refinement algorithm is small (less than 20), and is to a large extent independent on the number of documents. The evaluation of all the various criterion functions presented in this paper at each refinement step can be implemented efficiently and bounded by a constant determined by the document that contains the maximum number of terms, thus the overall amount of time required to compute a  $k$ -way clustering solution is  $O(kN)$ .

## 5 Experimental Results

We experimentally evaluated the performance of the various topic-driven clustering schemes, compared with the corresponding unsupervised and supervised clustering schemes on five datasets, and studied various issues associated with our topic-driven clustering schemes. In the rest of this section we first describe the various datasets and our experimental methodology, followed by a description of the experimental results.

### 5.1 Document Collections

In our experiments, we used a total of five different datasets, whose general characteristics are summarized in Table 2. The smallest of these datasets contained 1,560 documents and the largest contained 2,838 documents. To ensure diversity in the datasets, we obtained them from different sources. For all datasets, we used a

stop-list to remove common words, and the words were stemmed using Porter’s suffix-stripping algorithm [18]. Moreover, any term that occurs in fewer than two documents was eliminated.

The datasets *trec6*, *trec7* and *trec8* were derived from the Financial Times Limited (FT) and the Los Angeles Times (LATimes) articles that are distributed as part of the TREC collection [24]. We used the queries of the ad hoc test from TREC-6 [24], TREC-7 [24] and TREC-8 [24] as the topic prototypes, and derived the datasets by including all the relevant document in FT and LATimes to particular queries. The queries that have fewer than 10 relevant documents were eliminated from the datasets. Each TREC query contains a title, a description and a narrative. The title usually contains 2-3 words as the key words. The description describes what are the contents of the relevant document briefly, and the narrative provides more detailed descriptions. Thus, we could use the titles, descriptions and narratives to form the topic prototypes of different levels of specificity. In particular, we used the titles to form short topics, titles and descriptions to form medium topics, and all three parts to form long topics.

The dataset *re1* is from Reuters-21578 text categorization test collection Distribution 1.0 [16], and contains the documents from 25 categories. For *re1*, we selected documents that have a single class label. Finally, the dataset *wap* are from the WebACE project [17, 12, 4, 5]. Each document corresponds to a web page listed in the subject hierarchy of Yahoo! [27]. The original sources for the *re1* and *wap* datasets do not contain sufficient information that we can derive as topics. Thus, we selected the median document of each class as the representer of the content and treated it as the topic prototype.

Table 2: Summary of datasets used to evaluate the various clustering criterion functions.

Dataset	Topic #	Source	# of Docs	# of terms	# of classes
trec6	301-350	FT & LATimes	2619	32790	38
trec7	351-400	FT & LATimes	2838	33963	45
trec8	401-450	FT & LATimes	2804	36347	43
re1		Reuters-21578 [16]	1657	3758	25
wap		WebACE [12]	1560	8460	20

## 5.2 Experimental Methodology and Metrics

For each one of the different datasets we obtained a  $k$ -way clustering solution that optimized the various topic-driven criterion functions shown in Table 1 and Equations 4.9 and 4.8, where  $k$  is the number of topics (classes) present in the dataset. In addition, we compared the three topic-driven schemes with the cor-

responding unsupervised and supervised schemes. In particular, for supervised schemes, we compared the various  $\mathcal{I}_1$  and  $\mathcal{I}_2$  topic-driven criterion functions with  $\mathcal{S}_{\mathcal{I}}$ , and the various  $\mathcal{E}_1$  topic-driven criterion functions with  $\mathcal{S}_{\mathcal{E}}$ . To illustrate whether the performance improvements gained by the various topic-driven schemes are due to the initialization with topic prototypes, we also performed another set of experiments using topics as the initial seeds but optimizing the various unsupervised criterion functions, which we will refer to as the *seed-based* scheme.

In addition, for each TREC dataset, we also compared the various topic-driven clustering schemes using long, medium, and short topics as the topic prototypes to evaluate the effectiveness of the proposed methods with topic prototypes of different levels of specificity.

The quality of a clustering solution was evaluated using the *entropy* measure that is based on how the various classes of documents are distributed within each cluster.

Given a particular cluster  $S_r$  of size  $n_r$ , the entropy of this cluster is defined to be

$$E(S_r) = -\frac{1}{\log q} \sum_{i=1}^q \frac{n_r^i}{n_r} \log \frac{n_r^i}{n_r},$$

where  $q$  is the number of classes in the dataset and  $n_r^i$  is the number of documents of the  $i$ th class that were assigned to the  $r$ th cluster. The entropy of the entire solution is defined to be the sum of the individual cluster entropies weighted according to the cluster size, *i.e.*,

$$Entropy = \sum_{r=1}^k \frac{n_r}{n} E(S_r).$$

A perfect clustering solution will be the one that leads to clusters that contain documents from only a single class, in which case the entropy will be zero. In general, the smaller the entropy values, the better the clustering solution is.

To eliminate any instances that a particular clustering solution for a particular criterion function got trapped into a bad local optimum, in all of our experiments we found ten different clustering solutions. As discussed in Section 4 each of these ten clustering solutions correspond to the best solution (in terms of the respective criterion function) out of ten different initial partitioning and refinement phases.

## 5.3 Comparison of the Various Clustering Schemes

The first set of experiments was focused on evaluating the quality of the clustering solutions produced by the three topic-driven schemes and the corresponding

unsupervised, supervised, and seed-based schemes for the  $\mathcal{I}_1$ ,  $\mathcal{I}_2$  and  $\mathcal{E}_1$  criterion functions. The long topics (including titles, descriptions, and narratives) were used for the *trec6*, *trec7*, and *trec8* datasets in this set of experiments. The  $\alpha$  values used in the various schemes were fixed for all the datasets. We will discuss how those values were determined in Section 5.5.

Table 3 shows the relative improvements of the various topic-driven schemes and the seed-based scheme over the corresponding unsupervised and supervised schemes averaged over all the five datasets.

The results in Table 3 show several trends. First, all the topic-driven schemes outperform the corresponding unsupervised and supervised schemes, and the overall best topic-driven scheme is the normalized scheme (combining unsupervised and supervised with normalization), which achieved the most improvements for all the three criterion functions. Second, in addition to the improvements made by initializing using topics as seeds, all the topic-driven schemes made further improvements for  $\mathcal{I}_1$  and  $\mathcal{I}_2$ , which showed that the observed improvements are not only because of good initializations, but also because of the good properties of the proposed topic-driven criterion functions. For  $\mathcal{E}_1$ , the normalized scheme made additional improvements but to a less extent. Finally,  $\mathcal{I}_1$  achieved the most improvements by applying topic-driven schemes.

Table 4 shows the more detailed results of this set of experiments on each dataset. All the entries in Table 4 are entropy values, except for the two columns under the unsupervised and seed-based methods labeled “CrFun”, where the entries are the criterion function values of the clustering solutions. Note that the entropy values in the supervised method column were achieved by the supervised criterion functions compared against the other schemes. The supervised criterion function is  $\mathcal{S}_{\mathcal{I}}$  for  $\mathcal{I}_1$  and  $\mathcal{I}_2$ , and  $\mathcal{S}_{\mathcal{E}}$  for  $\mathcal{E}_1$ . The entries that are bold-faced correspond to the methods that perform the best for a particular dataset and criterion function.

A number of observations can be made by analyzing the results in Table 4. First, for most of the cases topic-driven schemes perform the best. The exception is the *wap* dataset, for which the seed-based scheme performed the best and the topic-driven schemes sometimes even performed worse than the unsupervised scheme. One of the differences between the *wap* dataset and the rest of the datasets is that the median documents used as topics are significantly longer than the topics used in other datasets. Since we did not perform any pruning on the median documents, they contain the terms that are not specific to the topic. As a result, the performances of the supervised schemes were much worse than other schemes and topic-driven schemes did not benefit from

incorporating these topics. Second, the two supervised schemes  $\mathcal{S}_{\mathcal{I}}$  and  $\mathcal{S}_{\mathcal{E}}$  perform similarly, which is not surprising as we have discussed the relationship between them in Section 4.1.2. Finally,  $\mathcal{E}_1$  benefits the most by using the topics as initial seeds. By looking at the criterion function values achieved by the unsupervised and seed-based schemes for  $\mathcal{E}_1$ , we can see that unlike  $\mathcal{I}_1$  and  $\mathcal{I}_2$ , using topics in the initialization process helped the optimization process to find a clustering solution with a better criterion function value for  $\mathcal{E}_1$ .

#### 5.4 Topic Prototypes of Different Levels of Specificity

The second set of experiments was focused on how the various topic-driven schemes perform with the topic prototypes of different levels of specificity. For each TREC dataset, we performed the same set experiments as in Section 5.3 with long, medium, and short topics. The results are shown in Table 5, in which all the entries are the entropy values of the clustering solutions obtained by the various schemes. Again, the entropy results for the weighted and normalized schemes were obtained with a fixed  $\alpha$  value for all the datasets. The entries that are bold-faced correspond to the methods that perform the best for a particular dataset and criterion function.

A number of observations can be made by analyzing the results in Table 4. First, the supervised scheme performs better as the topics become more specific for all the datasets. Second, for most of the cases topic-driven schemes perform the best. Finally, overall the various topic-driven schemes performed similarly with the topic prototypes of different levels of specificity, despite the fact that the short and medium topics alone (used in the supervised scheme) perform much worse than the long topics, which shows that the proposed topic-driven schemes are effective with the topic prototypes of different levels of specificity.

#### 5.5 Parameter Sensitivity

In this section, we present the results of the parameter study on  $\alpha$  for the weighted and normalized schemes, and show how we determined the  $\alpha$  values that were used to produce clustering solutions shown in Table 4. The purpose of this study is two-fold: (1) to see whether there is a range of  $\alpha$  values that can perform well for most of the datasets; (2) to see which scheme is better by comparing the dynamic range and how sensitive the two schemes are to the change of  $\alpha$  values. In particular, we tested the two schemes with  $\alpha = 0.1$  to 0.9 with an increment of 0.1 on the five datasets for all three criterion functions. Note that the two combined schemes emphasize more on the supervised component

Table 3: Average relative improvements of the various topic-driven schemes over the unsupervised and supervised schemes.

CrFun	Seed-based Scheme			Topic-driven Schemes								
	$\mathcal{E}_1$	$\mathcal{I}_1$	$\mathcal{I}_2$	Hybrid		Weighted			Normalized			
				$H(\mathcal{I}_1)$	$H(\mathcal{I}_2)$	$M_1(\mathcal{E}_1)$	$M_1(\mathcal{I}_1)$	$M_1(\mathcal{I}_2)$	$M_2(\mathcal{E}_1)$	$M_2(\mathcal{I}_1)$	$M_2(\mathcal{I}_2)$	
Unsupervised	6%	3%	5%	15%	9%	5%	26%	11%	9%	26%	12%	
Supervised	23%	4%	26%	16%	30%	22%	27%	31%	25%	27%	32%	

Table 4: Comparison of the clustering solutions obtained by the various clustering methods.

trec6									
CrFun	Unsupervised Method			Supervised Methods	Seed-based Method		Topic-driven Methods		
	CrFun	Entropy			CrFun	Entropy	$H()$	$M_1()$	$M_2()$
$\mathcal{E}_1$	3.98	0.238		0.281	3.96	0.210	0.215	<b>0.193</b>	
$\mathcal{I}_1$	4.37	0.275		0.283	4.32	0.268	0.238	<b>0.177</b>	
$\mathcal{I}_2$	1.01	0.208		0.283	1.01	0.196	<b>0.147</b>	0.178	0.162

trec7									
CrFun	Unsupervised Method			Supervised Methods	Seed-based Method		Topic-driven Methods		
	CrFun	Entropy			CrFun	Entropy	$H()$	$M_1()$	$M_2()$
$\mathcal{E}_1$	4.76	0.261		0.305	4.76	0.239	0.249	<b>0.228</b>	
$\mathcal{I}_1$	4.67	0.322		0.307	4.59	0.334	0.264	<b>0.217</b>	
$\mathcal{I}_2$	1.09	0.227		0.307	1.09	0.235	0.215	<b>0.207</b>	

trec8									
CrFun	Unsupervised Method			Supervised Methods	Seed-based Method		Topic-driven Methods		
	CrFun	Entropy			CrFun	Entropy	$H()$	$M_1()$	$M_2()$
$\mathcal{E}_1$	4.51	0.201		0.277	4.50	0.215	0.196	<b>0.185</b>	
$\mathcal{I}_1$	4.19	0.270		0.275	4.16	0.278	0.252	<b>0.175</b>	
$\mathcal{I}_2$	1.04	0.208		0.275	1.03	0.194	0.188	<b>0.160</b>	

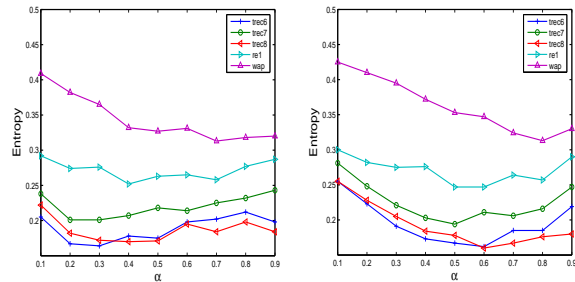
rel									
CrFun	Unsupervised Method			Supervised Methods	Seed-based Method		Topic-driven Methods		
	CrFun	Entropy			CrFun	Entropy	$H()$	$M_1()$	$M_2()$
$\mathcal{E}_1$	1.94	0.297		0.315	1.93	<b>0.273</b>	0.277	0.282	
$\mathcal{I}_1$	2.66	0.365		0.311	2.59	0.308	0.288	<b>0.252</b>	
$\mathcal{I}_2$	6.31	0.290		0.311	6.29	0.265	0.268	<b>0.247</b>	

wap									
CrFun	Unsupervised Method			Supervised Methods	Seed-based Method		Topic-driven Methods		
	CrFun	Entropy			CrFun	Entropy	$H()$	$M_1()$	$M_2()$
$\mathcal{E}_1$	1.75	0.331		0.442	1.76	<b>0.307</b>	0.327	0.337	
$\mathcal{I}_1$	1.65	0.372		0.454	1.65	0.355	<b>0.306</b>	0.359	
$\mathcal{I}_2$	4.69	0.327		0.454	4.66	<b>0.306</b>	0.342	0.332	

with a smaller  $\alpha$  value, and emphasize more on the unsupervised component with a larger  $\alpha$  value. The two combined schemes become the supervised scheme and the unsupervised scheme with  $\alpha = 0$  and 1, respectively.

The results for  $\mathcal{I}_1$  and  $\mathcal{I}_2$  are similar and we only show the results for  $\mathcal{I}_2$ . In Figure 1, we plot the entropy values obtained by the weighted and normalized schemes against the  $\alpha$  values for all the datasets in a) and b), respectively. We can see that at  $\alpha = 0.4$  and  $\alpha = 0.6$ , most of the datasets reached the best entropy value (or close to the best entropy value) for the weighted and normalized schemes, respectively. We also tested the same  $\alpha$  values on the same dataset with long, medium, and short topics. Figure 2 shows such a plot of the entropy values obtained by the weighted and normalized



a) Weighted Scheme      b) Normalized Scheme

Figure 1: Entropy values obtained by the weighted and normalized schemes with  $\mathcal{I}_2$  for all the datasets.

Table 5: Comparison of the clustering solutions obtained by the various clustering methods with long, medium, and short topics.

trec6							
Topic Type	CrFun	Unsupervised Method	Supervised Methods	Seed-based Method	Topic-driven Methods		
					$H()$	$M_1()$	$M_2()$
long	$\mathcal{E}_1$		0.281	0.210	0.215	<b>0.193</b>	
medium	$\mathcal{E}_1$	0.238	0.298	0.206	0.220	<b>0.191</b>	
short	$\mathcal{E}_1$		0.323	<b>0.211</b>	0.215	0.216	
long	$\mathcal{I}_1$		0.283	0.268	0.238	<b>0.177</b>	0.180
medium	$\mathcal{I}_1$	0.275	0.299	0.269	0.246	0.199	<b>0.198</b>
short	$\mathcal{I}_1$		0.323	0.279	0.314	0.222	<b>0.205</b>
long	$\mathcal{I}_2$		0.283	0.196	<b>0.147</b>	0.178	0.162
medium	$\mathcal{I}_2$	0.208	0.299	0.205	<b>0.157</b>	0.161	0.181
short	$\mathcal{I}_2$		0.323	0.208	0.186	<b>0.174</b>	0.190

trec7							
Topic Type	CrFun	Unsupervised Method	Supervised Methods	Seed-based Method	Topic-driven Methods		
					$H()$	$M_1()$	$M_2()$
long	$\mathcal{E}_1$		0.305	0.239	0.249	<b>0.228</b>	
medium	$\mathcal{E}_1$	0.261	0.334	0.244	0.239	<b>0.227</b>	
short	$\mathcal{E}_1$		0.371	0.242	<b>0.241</b>	0.247	
long	$\mathcal{I}_1$		0.307	0.334	0.264	<b>0.217</b>	0.219
medium	$\mathcal{I}_1$	0.322	0.332	0.318	0.279	<b>0.219</b>	0.221
short	$\mathcal{I}_1$		0.370	0.331	0.295	<b>0.247</b>	0.248
long	$\mathcal{I}_2$		0.307	0.235	0.215	<b>0.207</b>	0.211
medium	$\mathcal{I}_2$	0.227	0.332	0.246	<b>0.202</b>	0.210	0.204
short	$\mathcal{I}_2$		0.371	0.233	<b>0.208</b>	0.224	0.228

trec8							
Topic Type	CrFun	Unsupervised Method	Supervised Methods	Seed-based Method	Topic-driven Methods		
					$H()$	$M_1()$	$M_2()$
long	$\mathcal{E}_1$		0.277	0.215	0.196	<b>0.185</b>	
medium	$\mathcal{E}_1$	0.201	0.292	0.200	0.201	<b>0.193</b>	
short	$\mathcal{E}_1$		0.303	<b>0.194</b>	0.217	0.189	
long	$\mathcal{I}_1$		0.275	0.278	0.252	0.186	<b>0.175</b>
medium	$\mathcal{I}_1$	0.270	0.291	0.265	0.260	<b>0.190</b>	0.191
short	$\mathcal{I}_1$		0.295	0.282	0.255	<b>0.185</b>	<b>0.185</b>
long	$\mathcal{I}_2$		0.275	0.194	0.188	0.170	<b>0.160</b>
medium	$\mathcal{I}_2$	0.208	0.291	0.196	0.182	<b>0.169</b>	0.172
short	$\mathcal{I}_2$		0.300	0.202	0.176	<b>0.171</b>	0.174

schemes against the  $\alpha$  values for the *trec8* dataset in a) and b), respectively. The results are similar for *trec6* and *trec7* as well. As shown in Figure 2, the results of long, medium, and short topics are very similar to one another. In Figures 1 and 2, the curves produced by the normalized scheme are smoother than those by the weighted scheme. However, the dynamic range of the weighted scheme is narrower around the  $\alpha$  value that achieved the best entropy value than that of the normalized scheme, which suggests that the weighted scheme can achieve relative good performance with a broader choice of  $\alpha$  values for  $\mathcal{I}_2$ .

Figure 3 shows the same plot generated by the weighted and normalized schemes with  $\mathcal{E}_1$  for all the datasets. Unlike  $\mathcal{I}_1$  and  $\mathcal{I}_2$ , the trend and the best  $\alpha$  value differ from dataset to dataset for  $\mathcal{E}_1$ , especially for the weighted scheme, which suggests that the problem of different scales has greater impacts on  $\mathcal{E}_1$  than  $\mathcal{I}_1$  and  $\mathcal{I}_2$ . Another difference between  $\mathcal{I}_2$  and  $\mathcal{E}_1$  is that for short topics, the normalized scheme tends to achieve

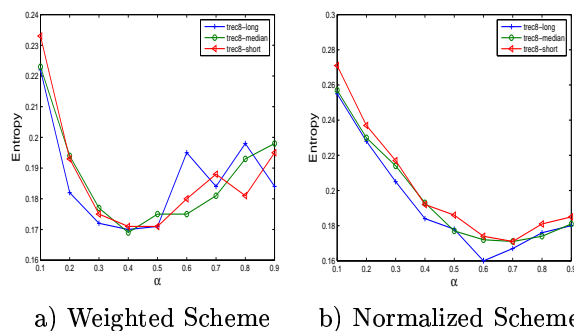


Figure 2: Entropy values obtained by the weighted and normalized schemes with  $\mathcal{I}_2$  for *trec8* with long, medium and short topics.

the best performance with a larger  $\alpha$  value as shown in Figure 4. Since there is no consistent trend for the weighted scheme with  $\mathcal{E}_1$ , we selected the  $\alpha$  value

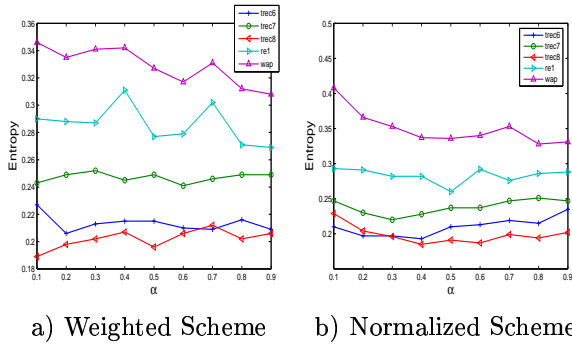


Figure 3: Entropy values obtained by the weighted and normalized schemes with  $\mathcal{E}_1$  for all the datasets.

Table 6: Selected  $\alpha$  values for the various combined schemes and criterion functions.

	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{E}_1$	$\mathcal{E}_1$ (short)
Weighted Scheme	0.4	0.4	0.5	0.5
Normalized Scheme	0.4	0.6	0.5	0.7

that achieved the best average entropy value. For the normalized scheme with  $\mathcal{E}_1$ , we determined the  $\alpha$  value in a similar fashion. The only difference is that we selected one  $\alpha$  value for short topics and another  $\alpha$  value for the rest of the cases.

In summary, the selected  $\alpha$  values that were used to produce the clustering solutions in Sections 5.3 and 5.4 are shown in Table 6.

Furthermore, we compared the performance of the two combined schemes with the fixed  $\alpha$  values with the best performance among all the tested  $\alpha$  values, and calculated the relative degradation on all the datasets. We show the box plots of the relative degradations for all the combined criterion functions in Figure 5.

A number of observations can be made by analyzing the results in Figure 5. First, for all the cases, the median relative degradation is lower than 2% except for  $M_1(\mathcal{E}_1)$ , which suggests that the fixed  $\alpha$  can perform well for most of the datasets. The poor performance of the fixed  $\alpha$  for  $M_1(\mathcal{E}_1)$  is consistent with that fact that the weighted scheme does not perform well with  $\mathcal{E}_1$  as shown in Figure 3. Second, the variance of the relative degradation of the normalized scheme is larger than that of the weighted scheme for  $\mathcal{I}_1$  and  $\mathcal{I}_2$ , which is consistent with the fact that the weighted scheme has a narrower dynamic range than the normalized scheme for  $\mathcal{I}_2$  as shown in Figure 1.

## 6 Concluding Remarks

In this paper, we defined the problem of *topic-driven clustering*, which organizes a document collection ac-

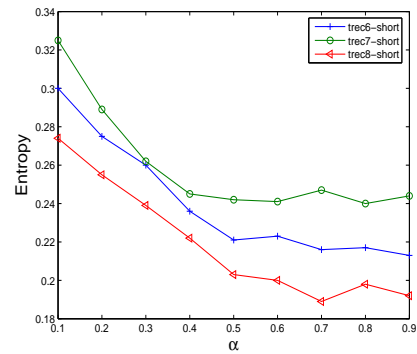


Figure 4: Entropy values obtained by the normalized scheme with  $\mathcal{E}_1$  and short topics.

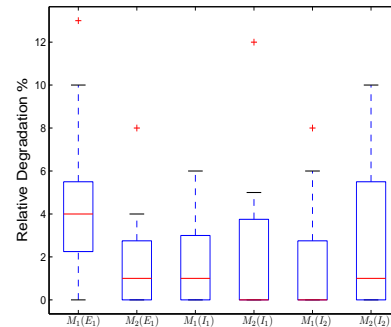


Figure 5: Relative performance of the selected  $\alpha$  for all the combined criterion functions.

ording to a given set of topics, such that the resultant clusters correspond to the given topics and the documents in the same cluster are similar to the cluster topic. We proposed three efficient topic-driven schemes that consider the similarity between the document to its topic and the relationship between the documents themselves simultaneously. Our experimental results showed that the proposed topic-driven schemes outperform the unsupervised and supervised schemes, which suggests that the proposed topic-driven schemes take advantages of both the unsupervised and supervised components. We also showed that the proposed topic-driven schemes perform well with topic prototypes of different levels of specificity.

## 7 Acknowledgments

We will like to thank Jack G. Conrad, Senior Research Scientist in Research & Development at Thomson Legal & Regulatory, for introducing us to the typical knowledge management environment of law firms and valuable discussions on the clustering requirements in law firms.

## References

- [1] Sugato Basu, Mikhail Bilenko, and Raymond J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proc. of the 10th Int'l Conference on Knowledge Discovery and Data Mining*, 2004.
- [2] Mikhail Bilenko, Sugato Basu, and Raymond J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proc. of 21th International Conference on Machine Learning (ICML-2004)*, 2004.
- [3] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Conference on Computational Learning Theory 11*, 1998.
- [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. *AI Review*, 11:365–391, 1999.
- [5] D. Boley, M. Gini, R. Gross, E.H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Partitioning-based clustering for web document categorization. *Decision Support Systems*, 27(3):329–341, 1999.
- [6] D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.
- [7] D.R. Cutting, J.O. Pedersen, D.R. Karger, and J.W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the ACM SIGIR*, pages pages 318–329, Copenhagen, 1992.
- [8] Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1/2):143–175, 2001.
- [9] Chris Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst Simon. Spectral min-max cut for graph partitioning and data clustering. Technical Report LBNL-47937, Lawrence Berkeley National Laboratory, University of California, Berkeley, CA, 2001.
- [10] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.
- [11] David Gondek and Thomas Hofmann. Non-redundant data clustering. In *Proc. of the fourth IEEE International Conference on Data Mining*, 2004.
- [12] E.H. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. WebACE: A web agent for document categorization and exploitation. In *Proc. of the 2nd International Conference on Autonomous Agents*, May 1998.
- [13] T. Joachims. *Advances in Kernel Methods: Support Vector Learning*, chapter Making large-Scale SVM Learning Practical. MIT-Press, 1999.
- [14] R. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. J. Wiley & Sons, New York, 1976.
- [15] Bjornar Larsen and Chinatsu Aone. Fast and effective text mining using linear-time document clustering. In *Proc. of the Fifth ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, pages 16–22, 1999.
- [16] D. D. Lewis. Reuters-21578 text categorization test collection distribution 1.0. <http://www.research.att.com/~lewis>, 1999.
- [17] J. Moore, E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, and B. Mobasher. Web page categorization and feature selection using association rule and principal component clustering. In *7th Workshop on Information Technologies and Systems*, Dec. 1997.
- [18] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
- [19] Jan Puzicha, Thomas Hofmann, and Joachim M. Buhmann. A theory of proximity based clustering: Structure detection by optimization. *PATREC: Pattern Recognition*, Pergamon Press, 33:617–634, 2000.
- [20] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.
- [21] K. Schloegel, G. Karypis, and V. Kumar. A new algorithm for multi-objective graph partitioning. In *Proceedings of Europar 1999*, September 1999.
- [22] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, 2000.
- [23] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2001.
- [24] TREC. Text REtrieval conference. <http://trec.nist.gov>, 1999.
- [25] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proc. of 18th International Conference on Machine Learning (ICML-2001)*, pages 577–584, 2001.
- [26] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. *Advances in Neural Information Processing Systems 15*, pages 505–512, 2003.
- [27] Yahoo! Yahoo! <http://www.yahoo.com>.
- [28] P. Yu. *Multiple-Criteria Decision Making: Concepts, Techniques, and Extensions*. Plenum Press, New York, 1965.
- [29] Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. Technical Report TR #01–40, Department of Computer Science, University of Minnesota, Minneapolis, MN, 2001. Available on the WWW at <http://cs.umn.edu/~karypis/publications>.
- [30] Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *Proc. of Int'l. Conf. on Information and Knowledge Management*, pages 515–524, 2002.
- [31] Y. Zhao and G. Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, 2004.