Category cluster discovery from distributed WWW directories

Mei-Ling Shyu [a], Choochart Haruechaiyasak [a], Shu-Ching Chen [b,1]

[a] *Department of Electrical and Computer Engineering*
University of Miami
Coral Gables, FL 33124, USA

[b] *Distributed Multimedia Information System Laboratory*
School of Computer Science, Florida International University
Miami, FL 33199, USA

Abstract

Due to the inherently distributed nature of many networks, including the Internet, information and knowledge are generated and organized independently by different groups of people. To discover and exploit all the knowledge from different sources, a method of knowledge integration is usually required. Considering the document category sets as information sources, we define a problem of information integration called category merging. The purpose of category merging is to automatically construct a unified category set which represents and exploits document information from several different sources. This merging process is based on the clustering concept where categories with similar characteristics are merged into the same cluster under certain distributed constraints. To evaluate the quality of the merged category set, we measure the precision and recall values under three classification methods, Naive Bayes, Vector Space Model, and K-Nearest Neighbor. In addition, we propose a performance measure called cluster entropy, which determines how well the categories from different sources are distributed over the resulting clusters. We perform the merging process by using the real data sets collected from three different Web directories. The results show that our merging process improves the classification performance over the non-merged approach and also provides a better representation for all categories from distributed directories.

Key words: Distributed information sources; Information integration; Cluster analysis; Web mining; Document classification

[1] Supported by National Science Foundation through grants CDA-9711582 and EIA-0220562

1 Introduction

With the amount of information growing at an exponential rate, the World Wide Web (WWW) is often referred to as the world's largest and fastest growing information source [20]. Unlike the traditional Information Retrieval System (IRS), WWW contains much larger amount of information than a typical IRS. Also due to the distributed nature of the Internet, the Web information is usually not well structured or organized. One of the most popular and successful approaches of organizing the information in the form of documents on WWW is to provide a directory containing a set of categories and classify the documents into these categories. At present, there are several Web portals such as *Yahoo!*, which provide its own directories for the users to browse and search for Web documents. The process of selecting the categories and organizing the documents is subjective to the groups of people in many different ways. Firstly, some categories in a directory may be different from those belonging to different directories. For example, a commercialized Web directory may include a category *shopping* for its targeted users, while an educational Web directory may include a category *science* in its directory. Secondly, the same Web document can be classified into different categories on different Web directories. For example, a document related to online stock trading may belong either to the category *business* in one directory or to the category *shopping* in another directory. Lastly, due to the huge amount of documents available on WWW, a document included in one directory may not be considered in a different directory.

The area of knowledge discovery from WWW is generally known as *Web mining* [4]. To alleviate the problem of information overload among the users, Web mining adopts the existing data mining tools in order to discover and exploit interesting and useful information from WWW. Some examples of Web mining techniques include analysis of user access patterns [12][18], Web document clustering [2][19][24], and classification [3][6][7]. Document classification or text categorization (as used in the information retrieval context) is the process of assigning a document to a predefined set of categories based on the document content. In order to classify a document into a category, a set of categories must be presented. To date, most of the research work in Web document classification use only a sample set of categories collected from a single document collection. Using only a single category set, however, might not provide a good view of the whole document domain, especially for the Web environment domain. As mentioned earlier, the amount of Web documents are enormous, and the process of collecting and organizing them is subjective. Thus, there is a need of integrating multiple category sets from distributed information sources.

In this paper, we propose a special problem of knowledge and ontology integration called *document category merging*. In general, the goal of knowledge and ontology integration is to construct an integrated knowledge base that exploits all the knowledge from several independent sources and has a good performance [1][11]. The need to merge the knowledge bases can arise when knowledge bases are acquired independently from interactions with several domain experts. As perspectives of different domain experts may differ, the knowledge bases constructed in this way will normally differ as well. If we consider the category set as a form of information source, then our document category merging can be considered as an integration problem whose goal is to capture and provide the best view of the categories by considering document collections from several different sources.

In order to merge multiple category sets into a unified one, we apply the clustering concept in which the categories with the similar characteristics are merged into the same cluster under certain distributed resource awareness constraints. Some of the typical clustering analysis issues and constraints include the limited number of clusters, the cluster size, i.e., number of items allowed within a cluster, and the cluster validity. Cluster validity is the process of evaluating and assessing the results of a clustering algorithm [5]. For category merging, we evaluate the clustering performance based on the precision and recall values given by some classification methods. In addition, we propose a performance measure called *cluster entropy*, which determines the quality of the clustering result under the distributed information environment. Intuitively, the cluster entropy of a clustering result is high when the number of clusters is large and each individual cluster contains the items belonging to different sources. The cluster entropy is discussed in more detail when the merging process is introduced in Section 3.

Two variations of the merging process called Merge1 and Merge2 are proposed by adopting the *Partitioning Around the Medoids* (*PAM*) clustering method [8]. Merge1 applies the clustering method without any restriction on the source distribution, while Merge2 is a restricted version of Merge1 with the inclusion of the distributed resource awareness constraint. To evaluate the performance of the merging process and to assess the quality of the merged category set, we measure the precision and recall values under three different classification methods, Naive Bayes, Vector Space Model, and K-Nearest Neighbor. We also use our proposed cluster entropy as another performance evaluation. To perform the experiments, we consider category sets collected from three different Web directories, *Yahoo!* [27], *Excite* [25] and *Open Directory Project (ODP)* [26]. The results show that our merging process not only generates a unified set of categories but also improves the classification performance, in terms of the averaged F_1 value (the average of precision and recall), by almost 50% over the non-merged approach.

The rest of the paper is organized as follows. In the next section, we present an overview of the classification approaches with the performance evaluation methods. In Section 3, we define the problem of category merging. Using the PAM clustering method, two variations of the merging process are presented. The cluster entropy is also explained in detail. In Section 4, experimental results using the real data sets are given. The paper concludes in Section 5.

2 Document classification and performance evaluation method

Document classification or text categorization is a well-studied research area in text mining and the information retrieval systems. Many algorithms have been proposed in the literatures. In this paper, we consider three approaches, Naive Bayes (NB), Vector Space Model (VSM), and K-Nearest Neighbor (KNN), for evaluating the performance of our merging process. The reviews of these classification methods are given in Sections 2.1, 2.2, and 2.3, respectively. The performance evaluation using the precision and recall values under the ranked-based thresholding strategy is explained in Section 2.4.

2.1 Naive Bayes (NB)

Naive Bayes (NB) classification method is based on a probabilistic model [9] [10]. The basic idea of the NB method for document classification is to use the joint probabilities of keywords and categories to estimate the probabilities of categories given a document. The NB classification makes an assumption of word independence, i.e., the conditional probability of a keyword given a category is assumed to be independent from the conditional probabilities of other keywords given that category. The process of classifying a document under NB is summarized as follows.

Given

- $C = \{c_1, c_2, \ldots, c_n\}$, a set of categories, where n is the number of categories, and
- $d = \{k_1, k_2, \ldots, k_m\}$, a test document, where k_i represents a keyword, and m is the total number of keywords in d,

the test document d is classified into category c_i where $P(c_i \mid d)$ is the maximum for all c_i, $1 \leq i \leq n$. $P(c_i \mid d)$ can be estimated under the independent assumption as follows [9].

$$P(c_i \mid d) = P(c_i) \times \frac{P(d \mid c_i)}{P(d)}, \tag{1}$$

where $P(d \mid c_i) = \prod_{j=1}^{m} P(k_j \mid c_i)$

2.2 Vector Space Model (VSM)

The Vector Space Model (VSM) is one of the classical clustering methods first proposed by [15]. This method has been successfully applied to many information retrieval systems including the well-known SMART system [14]. VSM assigns the attributes (keywords in this context) into n dimensional space, where n is the number of the attributes. The process of selecting and reducing the dimension n is called the *feature selection* [23]. After applying a feature selection technique, each document can be represented by an n dimensional vector called a *document vector*. Each position in a document vector is the value of the term frequency multiplied by the inverse of the document frequency, i.e., *tf-idf* [16]. For the document classification problem, we have some predefined set of categories, where each can also be represented by an n-dimensional vector called a category vector.

To classify a document into one of the categories, the document vector is compared with all category vectors using a similarity metric. The document is classified into the category where the similarity measure is the highest among all categories. Several approaches for calculating the similarity measures between documents and categories have been proposed [13]. Two types of measures have been widely used. The first is the distance metric (representing dissimilarity) such as *Euclidean distance*. The second type is similarity measures such as *cosine* and *dice* coefficients. In this paper, the cosine coefficient is used to calculate the similarity measures between documents and categories. The calculation of the cosine coefficient is given below [13].

$$COSINE(\vec{f}_i, \vec{g}_j) = \frac{\sum_{k=1}^{n}(f_{i,k} \times g_{j,k})}{\sqrt{\sum_{k=1}^{n} f_{i,k}^2 \times \sum_{k=1}^{n} g_{j,k}^2}}, \tag{2}$$

where

- $\vec{f}_i \in F$, F is a set of document vectors with n dimensions,
- $\vec{g}_i \in G$, G is a set of category vectors with n dimensions, and
- n represents the number of keywords.

2.3 K-Nearest Neighbor (KNN)

The K-Nearest Neighbor classification is a well-known statistical learning approach [22]. The process of the KNN method is similar to the VSM approach in which all the training and the test documents need to be modeled as n-dimensional vectors, where n is the number of keywords. In this paper, we use the same *tf-idf* strategy as applied in the VSM method to construct the document vectors. The KNN method is an instance-based learning method where each test document is compared against all the training documents to find the K nearest neighbors. Then the similarity measures between the test document to all K neighbors are summed up to give the weights for the corresponding categories. If several of the K nearest neighbors share the same category, then the similarity measures are added together as the weight for that category and the resulting weighted sum is used as the likelihood score of that category with respect to the test document. By sorting the scores of the categories, a ranked list is obtained for the test document to measure the performance evaluation by the precision and recall values. We use the cosine coefficient (given in Equation 2) as the similarity measure.

2.4 Classification performance evaluation

To evaluate the performance of a classification method, the commonly used measures are the precision and recall values. Precision is the ratio of the number of documents correctly classified into a category over the total number of documents being tested. Recall is the ratio of the number of documents correctly classified into a category over the total number of documents belonging to that category. Using the combination of the precision and recall, the F_1 measure which is the harmonic average of precision and recall can be defined as follows [22].

$$F_1 = \frac{2 \times (precision \times recall)}{precision + recall} \tag{3}$$

We use the ranked-based thresholding strategy (Rcut) in representing the classification results [21]. In general, the Rcut thresholding strategy is done by setting a threshold t whose value is between one and the maximum number of categories. In this paper, we will use the F_1 measure by setting the Rcut threshold t value to one, since each document from our data sets belongs to only one category.

3 Category merging process

In this section, we describe the category merging process with the performance evaluation method. Before we proceed, a definition of category merging is given as follows.

Definition 1 *Document category merging is a process of integrating two or more sets of categories from independently organized document collections into a unified set of categories. The results of the merging process are clusters of categories, such that when applying a classification method, its performance is improved over the non-merged category set.*

Next, an overview of the Partitioning Around the Medoids (PAM) clustering method with two variations of merging process is presented in Section 3.1. In Section 3.2, the cluster entropy, which is used as another performance evaluation in our experiments, is defined.

3.1 Merging process using Partitioning Around the Medoids (PAM)

The Partitioning Around the Medoids (PAM) can be viewed as a variation of the well-known *Nearest-Neighbor* (*NN*) clustering algorithm [8]. The idea of the *PAM* clustering method is as follows. In order to obtain k clusters, the method selects k objects (which are called the representative objects, or k-medoids) from the data set. The corresponding clusters are then found by assigning each remaining object to the nearest representation object based on the similarity or distance measures. In this paper, we adopt the PAM clustering method using the cosine coefficient in our merging process. The objects under consideration are categories from different directories. Since the clustering method is applied to the distributed information sources, therefore some distributed constraints must be included during the process. We propose two variations of the merging process, Merge1 and Merge2, which are explained as follows.

- Merge1: By fixing the number of the clusters equal to k, this approach selects k representative categories as the startups. Given a pre-merged category set, each category can be represented by a vector containing either the *tf-idf* values for the VSM and KNN classification methods, or the keyword probabilistic values for the NB method. Using these category representatives, the entropy values for each category can be calculated and used as the heuristic in selecting the k medoids. The categories whose entropy values are the k highest are picked as the startup categories. Next, the rest of the categories are clustered into one of the k medoids, by comparing the similarity measures. The only restriction under the Merge1 approach is the cluster size limit or the number of categories to be merged into the

same cluster. The cluster size limit is set to equal to the total number of pre-merged categories divided by the number of clusters allowed. This constraint is used to balance the cluster size to be evenly distributed.

- Merge2: This approach is similar to Merge1 except it includes the distributed constraint during the clustering process. This distributed constraint sets a higher priority to a category to be merged to a cluster if the category is originally from a different source (directory) from the medoid, i.e., the cluster representative. Since the bias or the overfitting problem might arise for the categories collected from the same source as opposed to the categories collected from different sources, the distributed constraint is used as a strategy to reduce this problem. Therefore, once the k medoids are selected, the rest of the categories are merged by the following rule. A test category is put into a cluster if the medoid is not from the same original source as the test category.

3.2 Cluster validity under the distributed information sources

In this section, we propose two types of cluster entropy measures to determine the quality of the clustering results. These measures are modified from the Shannon's entropy [17] to assess the clustering results under distributed information environment. The first measure *Entropy1* is based on the number of items belonging to each cluster without considering the item sources. In the second type *Entropy2*, the items are first identified by their distributed sources and the entropy calculation is based on these item sources. Therefore, *Entropy1* may be used to compare the clustering results between the Merge1 and Merge2 approaches without a bias on the distributed source constraint, and *Entropy2* may be used to assess the clustering quality under the distributed information sources. The formal definitions of these total cluster entropy measures are given as follows.

Definition 2 *Given the following preliminaries,*

- *m distributed information sources, $\{S_1, S_2, \ldots, S_m\}$, S_i contains a set of items, $\{s_1^i, s_2^i, \ldots, s_{|S_i|}^i\}$, where $\mid S_i \mid$ denotes the number of items in S_i,*
- *a clustering result of n clusters, $\{C_1, C_2, \ldots, C_n\}$, C_j contains a set of items, $\{c_1^j, c_2^j, \ldots, c_{|C_j|}^j\}$, where $\mid C_j \mid$ denotes the number of items in C_j, $c_k^j \in S_i, 1 \leq i \leq m, 1 \leq k \leq \mid C_j \mid$, and*
- *let $\mid N_j \mid$ be the number of information sources found in C_j,*

the cluster entropy measures, Entropy1 and Entropy2 are defined as follows.

$$Entropy1 = \sum_{j=1}^{n} \mid C_j \mid \times (-\frac{1}{\mid C_j \mid} log \frac{1}{\mid C_j \mid})$$
$$= \sum_{j=1}^{n} (-log \frac{1}{\mid C_j \mid}) \quad (4)$$
$$Entropy2 = \sum_{j=1}^{n} \mid N_j \mid \times (-\frac{1}{\mid N_j \mid} log \frac{1}{\mid N_j \mid})$$
$$= \sum_{j=1}^{n} (-log \frac{1}{\mid N_j \mid}) \quad (5)$$

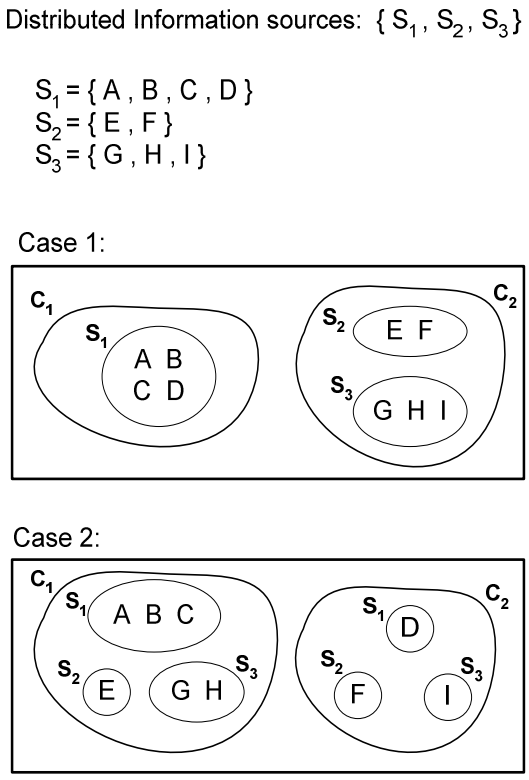

Fig. 1. Example of calculating cluster entropy measures.

Example 1 We show a calculation example of *Entropy1* and *Entropy2* from the clustering results. The distributed information sources with the item information are shown in Fig. 1. Two clustering results are shown in Case 1 and Case 2. For example, in Case 1, the clustering result contains 2 clusters, C_1 and C_2. C_1 contains 4 items, A, B, C, and D which all belong to S_1, therefore $\mid C_1 \mid$ is equal to 4 and $\mid N_1 \mid$ is equal to 1. C_2 contains 5 items, E, F, G, H, and I, where E and F belong to S_2 and G, H, and I belong to S_3, therefore $\mid C_2 \mid$ is equal to 5 and $\mid N_2 \mid$ is equal to 2. *Entropy1* and *Entropy2* for both cases are calculated as follows.

- Case 1:

$$Entropy1 = (-log\frac{1}{4}) + (-log\frac{1}{5}) \approx 4.32$$
$$Entropy2 = (-log\frac{1}{1}) + (-log\frac{1}{2}) \approx 1.00$$

- Case 2:

$$Entropy1 = (-log\frac{1}{6}) + (-log\frac{1}{3}) \approx 4.17$$
$$Entropy2 = (-log\frac{1}{3}) + (-log\frac{1}{3}) \approx 3.17$$

From the result, Entropy1 of Case 1 is higher than that of Case 2. This is due to the fact that the items in Case 1 are more evenly distributed than those of Case 2. Since the resulting clusters of Case 2 contain items from more different sources than those of Case 1, Entropy2 of Case 2 is higer than that of Case 1. Therefore, Entropy1 measures how well the items are evenly distributed among the clusters and Entropy2 measures how well the item sources are distributed over the clusters.

For the category merging process, the performance of the clustering result also depends on the classification performance. Thus, we combine the cluster entropy with the F_1 measure by applying the multiplication, e.g., (F_1 x Entropy1). This combined measure is used to evaluate the performance of the proposed merging process in the next section.

4 Experiments and results

In this section, we first provide the descriptions of the data sets used for performing our experiments. Then, the experimental results and discussions are presented.

4.1 Experimental data sets

Experiments using the predefined categories and the document sets collected from three Web directories, *Yahoo!* [27], *Open Directory Project – ODP* [26], and *Excite* [25], are conducted. In our experiments, we only consider documents in English and ignore all the other non-English documents. Therefore, the categories, World and Regional, are excluded from our experimental data sets.

Table 1
Predefined category sets from three Web directories.

Yahoo!	Excite	ODP
Arts & Humanities	Autos	Arts
Business & Economy	Computers	Business
Computers & Internet	Entertainment	Computers
Education	Games	Games
Entertainment	Health	Health
Government	Home & Real Estate	Home
Health	Investing	Kids and Teens
News & Media	Lifestyle	News
Recreation & Sports	Music	Recreation
Science	Relationships	Science
Social Science	Sports	Shopping
Society & Culture	Travel	Society
		Sports

Table 1 shows the selected categories from all Web directories, where the total number of categories is 37. Based on these predefined categories, we collect approximately 9,000 documents from each of the Web directories as the training and test sets. To avoid the problem of over-fitting the data when performing the experiments, we randomly select two-third of the document sets as the training set and one-third as the test set.

Considering only the training data sets of three Web directories, we extract and select some top-ranked keywords from each directory using the *document frequency* (*DF*) keyword selection technique [23]. We set the number of keywords from each directory to be equal to the number of categories multiplied by 100. Therefore, for Yahoo! and Excite data sets which have 12 categories, the number of selected keywords is 1,200. For the ODP data set, the number of selected keyword is 1,300. Next, the keywords from these three Web directories are combined into a single set. Some of the keywords appear in more than one Web directory, but we only consider one instance for each of these. The total number of all distinct keywords is 1,757. After this step, both the training and test document sets collected from all three directories are cleaned by considering only those distinct keywords.

Once the training and test document sets are preprocessed, we apply the merging processes, Merge1 and Merge2, under three classification methods, Naive Bayes, Vector Space Model, and K-Nearest Neighbor. The performance results under *Entropy1* and *Entropy2* are shown in Fig. 2 and Fig. 3, respectively.

As can be observed from Fig. 2 and Fig. 3, we execute our merging processes by varying the number of clusters from 2 to 25. The purpose of varying the numbers of clusters is to observe and determine the best possible clustering result. Our evaluation performance criteria depend on both the precision and recall of the classification method being tested and also the cluster entropy. Therefore, we measure the results in term of F_1 value (measured when the Rcut thresholding value is equal to 1) multiplied by our proposed cluster entropy measures, *Entropy1* and *Entropy2*. We refer to this combination as $(F_1 \times Entropy1)$ and $(F_1 \times Entropy2)$, respectively. From Fig. 2 and 3, we denote the Naive Bayes, Vector Space Model, and K-Nearest Neighbor classification methods by NB, VSM and KNN, respectively. The numbers 1 and 2 after the classification abbreviations denote the merging approaches, Merge1 and Merge2, respectively. For example, NB(1) denotes the Naive Bayes method using the Merge1 approach.

From Fig. 2 and 3, it can be observed that using either Merge1 or Merge2, the KNN classification method yields higher $(F_1 \times Entropy1)$ and $(F_1 \times Entropy2)$ values for most of the cluster numbers, whereas NB and VSM yield a relatively worse performance. Under $(F_1 \times Entropy1)$, the Merge1 approach has the maximum performance values when the number of clusters is 12 for all classification methods. Under $(F_1 \times Entropy2)$, the Merge2 approach has the peak performance when the number of the clusters is 13 for all three classification methods. Next, we use these peak performance information to compare the precision and recall values among the non-merged, Merge1, and Merge2 approaches.

To see the improvement of the merged category set over the non-merged one, we plot three precision vs. recall graphs showing the performance under three classification methods. The first one compares Merge1 to the non-merged approach. The second one compares Merge2 to the non-merged approach. The third one compares between the Merge1 and Merge2 approaches. The comparison graphs are shown in Fig. 4, Fig. 5, and Fig. 6, respectively. Note that for both Merge1 and Merge2, we only consider the clustering results when the maximum performance occurs as seen from Fig. 2 and 3, respectively. Similarly, we use the numbers 1 and 2 after the classification abbreviation to denote Merge1 and Merge2, respectively, and denote the non-merged approach with the word *non*.

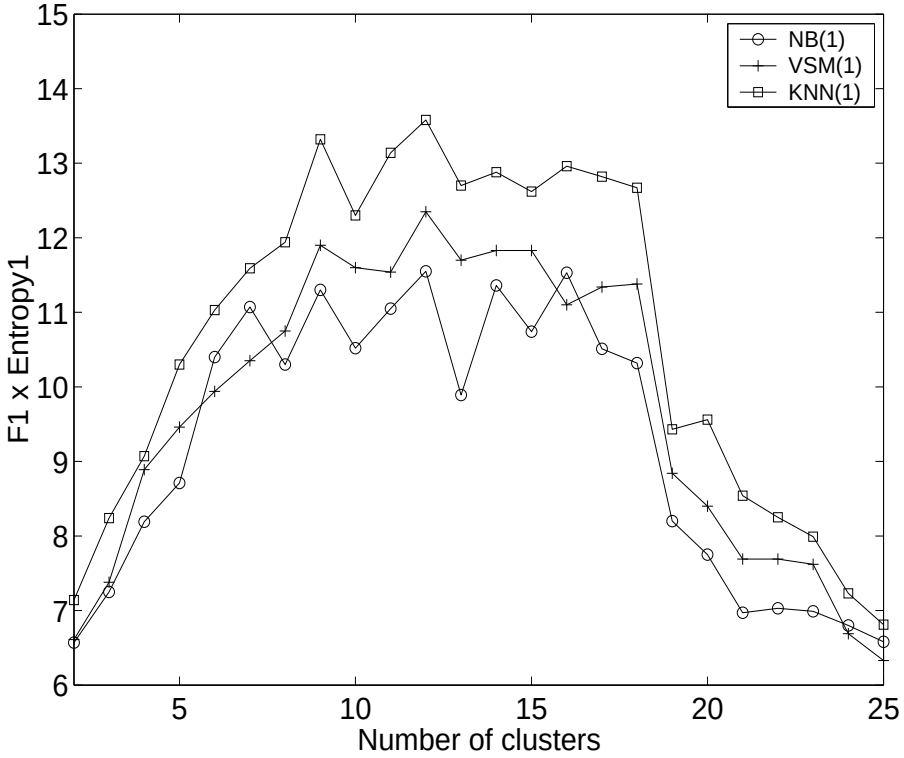

Fig. 2. Performance evaluation of merging processes under Entropy1.

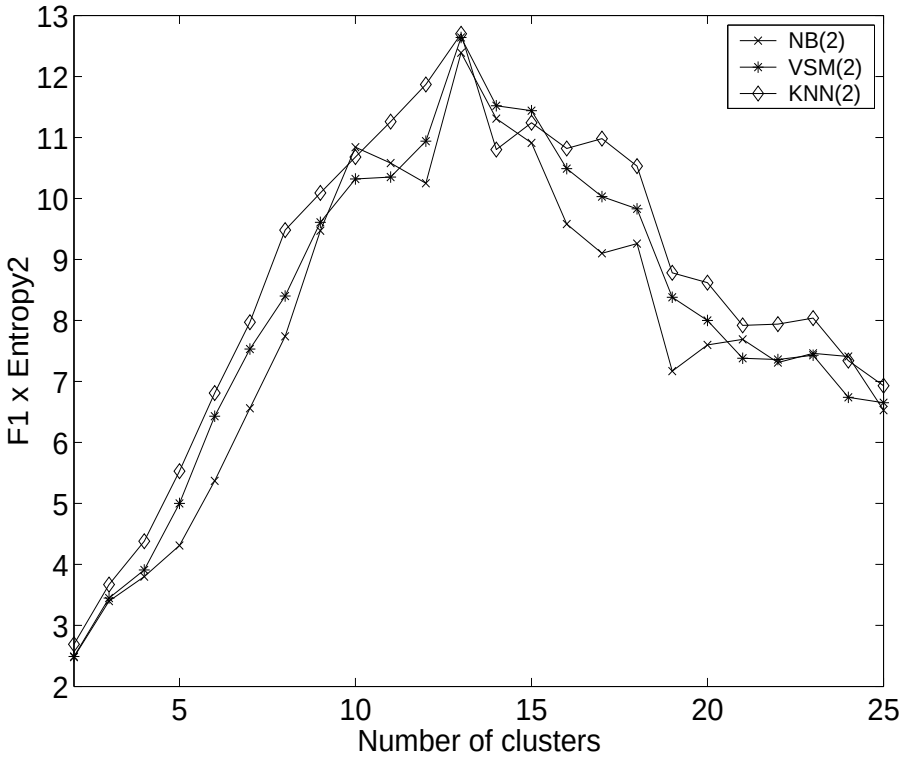

Fig. 3. Performance evaluation of merging processes under Entropy2.

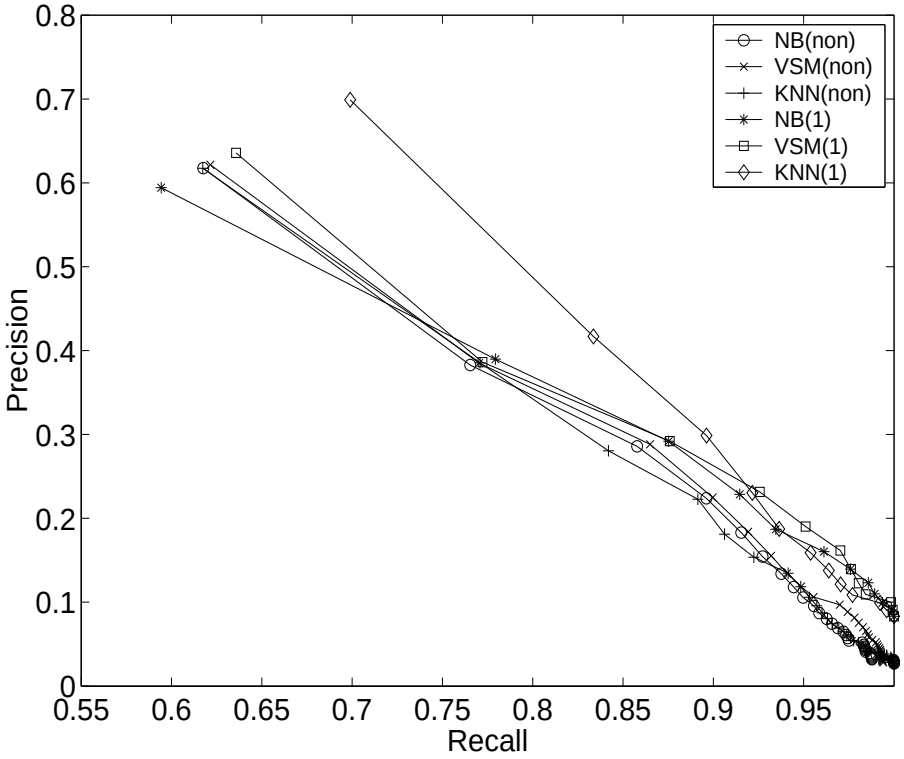

Fig. 4. Comparison between the Merge1 and non-merged approaches.

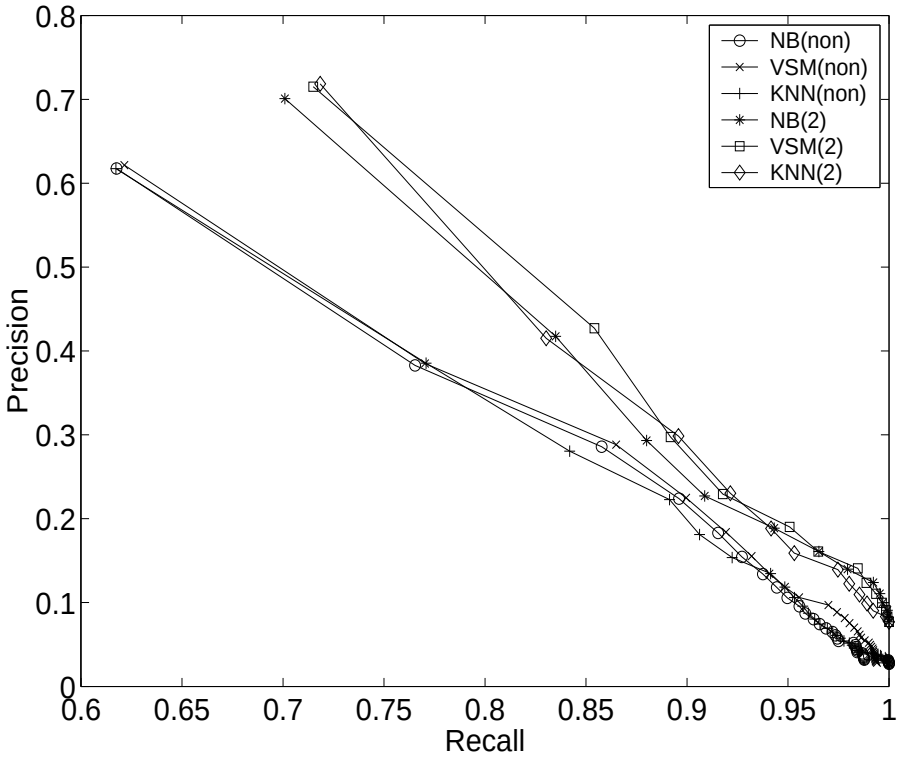

Fig. 5. Comparison between the Merge2 and non-merged approaches.

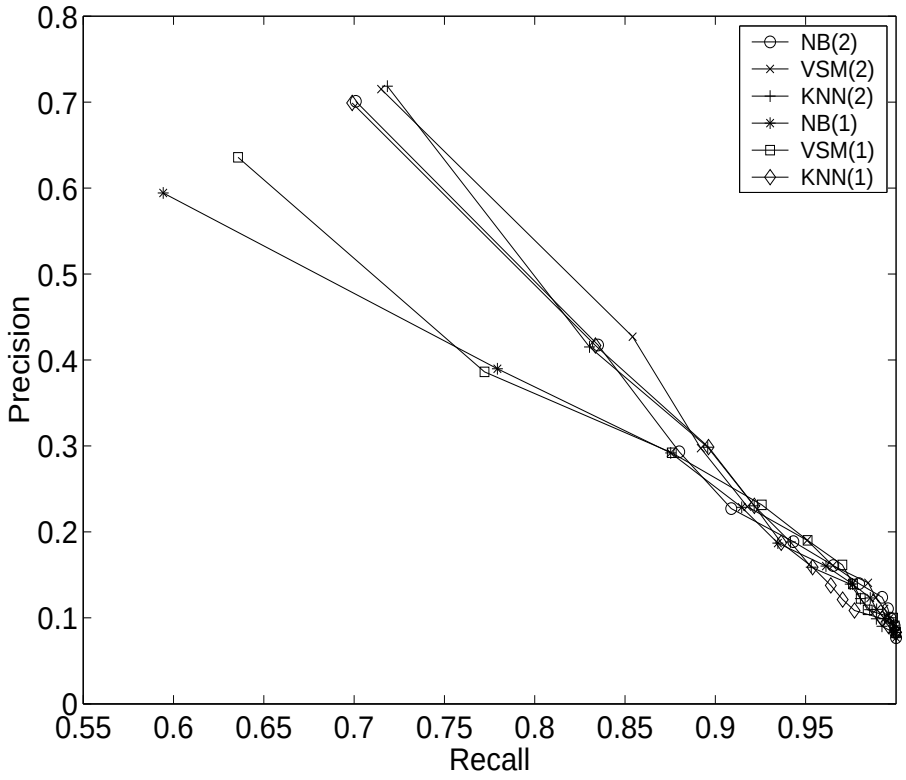

Fig. 6. Comparison between the Merge1 and Merge2 approaches.

From Fig. 4, it can be observed that by using Merge1 to find the category clusters, the classification performance does not improve much, except for the KNN method. Therefore, applying the clustering without any restriction may not guarantee a significantly improvement in the classification performance. In Fig. 5, the comparison between Merge2 and the non-merged approaches is shown. It can be immediately seen that the classification performance of Merge2 is significantly better than of the non-merged approach, which demonstrates the effectiveness of the distributed constraint imposed during the clustering process. Finally, Fig. 6 shows that the Merge2 approach yields a better classification performance than the Merge1 approach.

Table 2 gives a summarized result of all three approaches by using the averaged F_1 values. From this table, it can be summarized that by using the Merge1 approach, the classification performance is improved over the non-merged approach by approximately 29%, 34%, and 42% for NB, VSM and KNN classification methods, respectively. By using the Merge2 approach, the classification performance is improved over the non-merged approach by approximately 46%, 47%, and 48% for NB, VSM and KNN classification methods, respectively. Therefore, it can be concluded that using Merge2 with the proposed distributed resource awareness constraint to cluster the categories into a unified set is an effective way to improve the classification performance.

Table 2
Averaged F_1 measures for the Merge1, Merge2, and non-merged approaches.

Classification and Merging Approach	Averaged F_1 value
NB (non)	0.2561
NB (1)	0.3307
NB (2)	0.3732
VSM (non)	0.2577
VSM (1)	0.3445
VSM (2)	0.3779
KNN (none)	0.2560
KNN (1)	0.3646
KNN (2)	0.3779

Table 3
The representative merged category set.

Cluster index	Cluster representation
1	Recreation, Sports, Travel
2	Society, Government, Lifestyle, Relationships
3	Science
4	Business, Economy, Home & Real Estate
5	Computers, Internet, Investing
6	Arts, Humanities, Entertainment, Music
7	News & Media
8	Health
9	Entertainment, Shopping
10	Games, Social Science
11	Education, Sports
12	Home, Society, Culture
13	Autos, Kid & Teens

Next, we present the clustering results by the category names. The maximum performance for the Merge2 approach occurs when the number of clusters is equal to 13 for all classification methods. Therefore, we use this clustering

result to construct the representative merged category set. In order to represent the name of each cluster, we combine all category names and ignore the duplicate ones. The result is shown in Table 3. As can be seen from Table 3, the merged category set showed the tendency of having clusters whose merged categories share similar names. For example, in category 1, the cluster represents the category of *Recreation, Sports, and Travel*. Therefore, in addition to the improvement in the classification performance, applying the category merging concept also provides a better view of all categories from distributed information sources.

5 Conclusion

In this paper, we defined a problem of information integration from distributed information sources called *document category merging*. Considering the category sets as information sources, the goal of category merging is to construct an integrated category set which exploits all the information and knowledge from several heterogeneous and distributed information sources. Our merging process is based on the concept of clustering method where categories with similar characteristics are merged together into the same cluster. We adopted and modified the Partitioning Around the Medoids (PAM) clustering method in our merging process. To evaluate the quality of the integrated category set, we measured and compared the F_1 value (a harmonic mean of precision and recall values) by performing three different classification methods, Naive Bayes (NB), Vector Space Model (VSM), and K-Nearest Neighbor (KNN). We also proposed another performance measures called cluster entropy, which measures how well the categories from heterogeneous information sources are distributed over the clusters. Under the precision and recall values, the results showed that using our proposed Merge2 merging process with the distributed resource awareness constraint, the performance gain, in term of the averaged F1 measure, was almost 50% over the non-merged approach. A final result of the merged category set showed the tendency of having clusters whose merged categories share similar names. Therefore, applying our category merging concept as a form of information integration provides a better view of all categories from distributed information sources, and also improves the classification performance.

References

[1] P. Brazdil, L. Torgo, Knowledge Acquisition via Knowledge Integration, in: B. Wielinga et al., eds., Current Trends in Artificial Intelligence, IOS Press, Amsterdam, 1990, pp.412–423.

[2] A. Broder, S. Glassman, M. Manasse, G. Zweig, Syntactic Clustering of the Web, in: Proc. of the Sixth Int. WWW Conf., 1997, pp. 391-404.

[3] S. Chakrabarti, B. Dom, P. Indyk, Enhanced hypertext categorization using hyperlinks, in: Proc. of the ACM SIGMOD Int. Conf. on Management of Data, 1998, pp. 307-318.

[4] R. Cooley, B. Mobasher, J. Srivastava, Web Mining: Information and Pattern Discovery on the World Wide Web, in: Proc. of the 9th IEEE Int. Conf. on Tools with Artificial Intelligence, 1997, pp. 558–567.

[5] M. Halkidi, Y. Batistakis, M. Vazirgiannis, Cluster Validity Methods: Part I, SIGMOD Record 31(2) (2002) 40–45.

[6] C. Haruechaiyasak, M.-L. Shyu, S.-C. Chen, X. Li, Web Document Classification Based on Fuzzy Association, in: Proc. of the 26th IEEE Int. Computer Software and Applications Conf., 2002, pp. 487–492,

[7] C. Haruechaiyasak, M.-L. Shyu, S.-C. Chen, Identifying Topics for Web Documents through Fuzzy Association Learning, in: International Journal of Computational Intelligence and Applications (IJCIA), Special Issue on Internet-Based Intelligent Systems, 2(3) (2002) 277-285.

[8] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons, 1990.

[9] D. Lewis, Naive Bayes at Forty: The Independence Assumption in Information Retrieval, in: Proc. of European Conf. on Machine Learning, 1998, pp. 4–15.

[10] A. McCallum, K. Nigam, A Comparison of Event Models for Naive Bayes Text Classification, in: Proc. of the AAAI-98 Workshop on Learning for Text Categorization, 1998, pp. 41–48.

[11] H. S. Pinto, A. G. Perez, J. P. Martins, Some Issues on Ontology Integration, in: Proc. of IJCAI Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends, 1999, pp. 7.1–7.12.

[12] J. Pitkow, P. Pirolli, Mining Longest Repeating Subsequences to Predict World Wide Web Surfing, in: Proc. USENIX Symp. on Internet Technologies and Systems, 1999, pp. 139–150.

[13] E. Rasmussen, Chapter 16: Clustering Algorithms, in: W. B. Frakes, R. Baeza-Yates, eds., Information Retrieval: Data Structures & Algorithms, Prentice Hall, 1992, pp. 419–442.

[14] G. Salton, ed., The SMART Retrieval System: Experiments in Automatic Document Processing, Prentice-Hall Series in Automatic Computation, Englewood Cliffs, NJ, 1971.

[15] G. Salton, A. Wong, C. S. Yang, A Vector-Space Model for Information Retrieval, Commun. ACM, 18(11) (1975) 613–620.

[16] G. Salton, C. Buckley, Term Weighting Approaches in Automatic Text Retrieval, Info. Proc. and Mgt. 24(5) (1988) 513–523.

[17] C. Shannon, A Mathematical Theory of Communication, Bell System Technical J., 27 (1948) 379–423, 623–656.

[18] M.-L. Shyu, S.-C. Chen, C. Haruechaiyasak, Mining User Access Behavior on the WWW, in: IEEE Int. Conf. on Syst., Man, and Cybernet., 2001, pp. 1717–1722.

[19] M.-L. Shyu, S.-C. Chen, C. Haruechaiyasak, C.-M. Shu, S.-T. Li, Disjoint Web Document Clustering and Management in Electronic Commerce, in: Seventh Int. Conf. on Distr. Multi. Syst., 2001, pp. 494–497.

[20] J. Wang, A Survey of Web Caching Schemes for the Internet, ACM Comp. Comm. Review, 29(5) (1999) 36–46.

[21] Y. Yang, A Study on Thresholding Strategies for Text Categorization, in: Proc. of the 24th ACM Int. Conf. on Research and Development in Information Retrieval, 2001, pp. 137–145.

[22] Y. Yang, An Evaluation of Statistical Approaches to Text Categorization, J. Info. Retrieval 1(1/2) (1999) 67–88.

[23] Y. Yang, J. P. Pedersen, A Comparative Study on Feature Selection in Text Categorization, in: Proc. of the Fourteenth Int. Conf. on Machine Learning, 1997, pp. 412–420.

[24] O. Zamir, O. Etzioni, Web Document Clustering: A Feasibility Demonstration, in: Proc. of the 21st Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, 1998, pp. 46–54.

[25] http://www.excite.com, Excite Directory.

[26] http://dmoz.org, Open Directory Project (ODP).

[27] http://www.yahoo.com, Yahoo! Web Search Directory.