Automatic Text Categorization and Its Application to Text Retrieval

Wai Lam[+] Miguel Ruiz[Λ] Padmini Srinivasan[*]

[+]Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong
Shatin
Hong Kong
wlam@se.cuhk.edu.hk

[Λ]Department of Computer Science
The University of Iowa
Iowa City, Iowa 52242
U.S.A.
mruiz@cs.uiowa.edu

[*]School of Library and Information Science
The University of Iowa
Iowa City, Iowa 52242
U.S.A.
padmini-srinivasan@uiowa.edu

Affiliation of Author

Wai Lam[1]

Department of Systems Engineering and Engineering Management

The Chinese University of Hong Kong

Shatin

Hong Kong

wlam@se.cuhk.edu.hk

Miguel Ruiz

Department of Computer Science

The University of Iowa

Iowa City, Iowa 52242

U.S.A.

mruiz@cs.uiowa.edu

Padmini Srinivasan

School of Library and Information Science

The University of Iowa

Iowa City, Iowa 52242

U.S.A.

padmini-srinivasan@uiowa.edu

[1]This work was partially supported by the CUHK Engineering Faculty Direct Grant 2050179.

Abstract

We develop an automatic text categorization approach and investigate its application to text retrieval. The categorization approach is derived from a combination of a learning paradigm known as instance-based learning and an advanced document retrieval technique known as retrieval feedback. We demonstrate the effectiveness of our categorization approach using two real-world document collections from the MEDLINE database. Next we investigate the application of automatic categorization to text retrieval. Our experiments clearly indicate that automatic categorization improves the retrieval performance compared with no categorization. We also demonstrate that the retrieval performance using automatic categorization achieves the same retrieval quality as the performance using manual categorization. Furthermore, detailed analysis of the retrieval performance on each individual test query is provided.

Index Terms: Text Categorization, Automatic Classification, Text Retrieval, Instance-Based Learning, Query Processing

1 Introduction

Text categorization has recently become an active research topic in the area of information retrieval. The objective of text categorization is to assign entries from a set of pre-specified categories to a document. A document here refers to a piece of text. Categories may be derived from a sparse classification scheme or from a large collection of very specific content identifiers. Categories may be expressed numerically or as phrases and individual words. Traditionally this categorization task is performed manually by domain experts. Each incoming document is read and comprehended by the expert and then it is assigned a number of categories chosen from the set of pre-specified categories. It is inevitable that a large amount of manual effort is required. For instance, the MEDLINE corpus, which consists of medical journal articles, requires considerable human resources to carry out categorization using a set of MeSH (Medical Subject Headings) categories [11].

A promising way to deal with this problem is to learn a categorization scheme automatically from training examples. Once the categorization scheme is learned, it can be used for classifying future documents. It involves issues commonly found in machine learning problems. Since a document may be assigned to more than one category, the scheme also requires the assignment of multiple categories.

There is a growing body of research addressing automatic text categorization. For instance, a probabilistic model, in the early work of Lewis [8], makes use of Bayesian independence classifiers for categorization. He mainly studies the effect of feature selection and clustering on the automatic categorization of newswire articles. Masand *et. al.* [10] adopt a memory-based reasoning strategy to classify news stories. After k best documents are retrieved, the weight of the associated categories are obtained by summing similarity scores from the near matches. Yang [20] develops a technique known as Expert Network. This network links the terms in a document with its categories and there is a weight on each link. Both approaches of Masand *et. al.* and Yang are similar to our approach in that these are based on variants of the nearest neighbor algorithm. However, they do not mention a model for parameter selection. Other methods such as decision trees [1], linear classifiers [9], context-sensitive learning [3], and learning by combining classifiers [7] have also been proposed. These approaches typically construct a classifier for each category and the categorization process becomes a binary decision

problem for the particular category. In contrast, our approach learns all the categories for a document at one time. More recently Lewis *et. al.* [9] compare three categorization algorithms: Rocchio's, Widrow-Hoff and Exponential Gradient on the Heart Disease subset of a MEDLINE test collection. Yang [21] also tests her Expert Network method on the same Heart Disease collection as well as a different MEDLINE test collection. We compare our categorization results to theirs in a later section.

All recent efforts on automatic text categorization have focused on the categorization task alone. One useful application for automatic categorization is to support effective text retrieval. Apart from studying the effectiveness of automatic categorization directly, the second objective of this paper is to investigate the application of this categorization process to text retrieval. In particular, we wish to study whether the automatically assigned categories will improve the retrieval performance compared with no categorization. We also investigate whether automatic categorization will improve, reduce or have no effect on the retrieval performance achieved using manual categorization. Furthermore, we analyze the retrieval performance on the basis of each individual test query to gain insight on the interaction of our automatic categorization and our text retrieval approaches.

This paper is organized in two parts: Part I focuses directly on the automatic categorization approach and Part II focuses on the application of categorization to text retrieval. For Part I, a description of the categorization approach is given in Section 2. The following section discusses different categorization quality metrics. This is followed by a section presenting the experimental results for automatic categorization on two document collections, namely, the HERSH [5] and the OHSUMED [6] test collections. For Part II, Section 5 presents the text retrieval approach based upon the automatic categorization approach. It is followed by a section describing a series of text retrieval experiments on the HERSH corpus. Finally Section 8 provides the conclusions of this paper.

Part I

2 A Description of the Categorization Approach

2.1 An Outline of the Approach

The basic components of the automatic categorization approach consists of two processes, namely the *category extraction process* and the *parameter selection process*. The category extraction process is responsible for extracting the appropriate categories for an input document. Central to this process is a *category learning model*. This model provides an algorithm to identify categories for a new document from a collection of existing pre-categorized document examples. We propose an approach derived from a combination of a machine learning technique known as *instance-based learning* and a text retrieval technique known as *retrieval feedback*. Retrieval feedback has been discussed in [2], [4], [13], [16], and [17]. It is a technique that is different from the traditional *relevance feedback* technique. Essentially, retrieval feedback supports a kind of automatic query refinement procedure which does not require manual relevance judgments from users as in traditional relevance feedback.

Many existing approaches build a separate classifier for each category. A new document is processed by each classifier to determine if the corresponding category is appropriate. In contrast, our approach operates at the document level. A *set* of categories is identified for a document in a single run. The category extraction process operates according to the category learning model. The learning model requires some operational parameters which will be selected in advance by the parameter selection process. This process also makes use of the same category learning model embedded in the category extraction process. We pose this parameter selection task as a simple optimization problem. Specifically the parameters are chosen so that the performance of the categorization process, measured by a metric, is optimized. We use a tuning set approach to achieve this task.

The interaction of all the components in our categorization approach is illustrated in Figure 1. For a given domain, we first invoke the parameter selection process to determine the appropriate parameter values. This step is carried out only once off-line at the beginning. After this step, we can determine the categories for a new document via the category extraction process which can be done efficiently

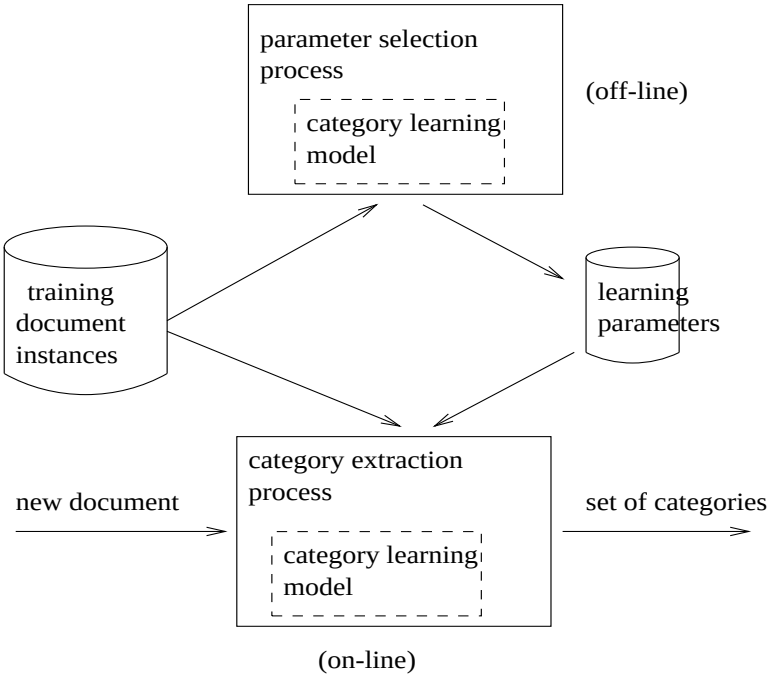


Figure 1: The Automatic Categorization Approach

on-line. The category learning model will be presented first since it is used in both processes. Then it is followed by a description of the parameter selection process and the category extraction process.

2.2 The Category Learning Model

Recall that the objective of this model is to select the appropriate categories for an input document. We adopt an extension of instance-based learning. There is a collection of pre-categorized documents used for training. Each document contains a free–text portion (typically the title and the abstract) and a set of categories that have been manually assigned. A document in the training collection is considered as an *instance* or *exemplar* represented by $T; C$ where T and C denote representations of the free–text and the categories of the document respectively.

We adopt the *vector space* technique as the central representation framework for our model. Thus T is a vector of terms:

$$T = (t_1, t_2, \ldots, t_p)$$

where p is the total number of unique terms in the collection's free–text domain and t_i is the weight reflecting the relative importance of the term i for characterizing the document. Standard automatic document indexing techniques in information retrieval can be employed to extract terms from a document. Typically a term can be a word or a phrase aside from common function words such as "the". Word stemming can be applied in the term extraction process. Our category learning model imposes no restriction on word stemming methods. Similarly C is a vector representing the categories assigned to the document.

$$C = (c_1, c_2, \ldots, c_q)$$

where c_i is the weight for the category i and q is the total number of unique categories.

A number of weighting schemes can be used for the vectors T and C. For instance, we can use the product of *term frequency* and *inverse document frequency* as the weight for the terms in T. Term frequency is the number of occurrences of a term in a document. Inverse document frequency is related to the rarity of a term in a document collection. More details of these two quantities can be found in [15]. For the vector C, we can use the *category frequency* as the weight. Usually the category frequency is binary. Both vectors are normalized before any further processing is done.

Let S denote an incoming document which needs to be categorized. Since the free–text of the document is available, we represent S by a vector of terms:

$$S = (s_1, s_2, \ldots, s_p)$$

where s_i is the weight for the term i in S. Here, the weighting scheme must be the same as the one used in the representation of T. The vector S should also be normalized.

As in instance-based learning, this document S is matched against each instance (*i.e.*, document) in the training collection according to a similarity function Δ. This function Δ produces a score for each training document instance. The higher the score, the higher is the similarity between S and a document instance. A simple and effective choice of this function is:

$$\Delta(S, T) = \sum_{i=1}^{p} s_i t_i$$

Note that both vectors S and T have been normalized before the calculation.

Based on this score, we rank the document instances in descending order. In instance-based learning, the categories associated with the most similar document (*i.e.,* the highest score) are the learned categories for S. Instead, in our approach, we gather the top N document instances to form a set Ψ (note that $|\Psi| = N$). Then, the categories associated with the documents in Ψ are further considered.

If we think of document S as a query, the set Ψ can be viewed as a set containing the N most relevant documents retrieved for the query S. In retrieval feedback, a query is automatically refined or expanded using the terms in Ψ. For our categorization task, we wish to find the appropriate categories for the query (*i.e.,* document S). Inspired by the retrieval feedback technique, we expand the query with categories selected from those associated with the documents in Ψ. To do this, a weight c_i' is first calculated for each category i associated with Ψ. This weight is computed by:

$$c_i' = \sum_{k=1}^{N} c_{ki} \tag{1}$$

where c_{ki} is the weight of category i in the k-th document in the set Ψ.

We rank the categories according to two criteria. First the categories are ranked in descending order by the number of documents in Ψ in which they occur. Next they are ranked in descending order of the computed weight c_i'. Finally the top M categories are extracted as the learned categories for the document S.

Note that N and M are the two parameters involved in the learning process. In the category extraction process, a selected value for each parameter is required. Instead of using arbitrary values, we propose a parameter selection process where good values for the parameters are determined in advance. This also allows the parameters to capture the characteristics of the document collection.

2.3 The Parameter Selection Process

The purpose of the parameter selection process is to determine suitable parameter values to be used in the category extraction process. We make use of a tuning set approach to achieve this. It is able to capture specific distinct characteristics of the document collection. This process is conducted off-line and needs to be performed only once for a given document collection.

Recall that we gather a set of documents with known categories as a training document collection. This training document collection is further partitioned into two parts. One part, denoted by Υ, is treated as the set of exemplars. The other part, denoted by Θ, is the tuning set. Each document in Θ is categorized based on the exemplars in Υ using the same categorization learning model described above. Since the correct categories (*i.e.* manually assigned categories) are available for the documents in Θ, we can measure the categorization performance by comparing the learned categories and the correct categories using a quality metric. This allows us to evaluate the categorization performance for a particular choice of parameters. We repeat this tuning process in a systematic way for different combination of parameters. The algorithm for this parameter selection process is summarized as follows:

1. choose an initial set of parameter values
2. for each document in Θ
    2.1 invoke the category learning model using the exemplar set Υ
3. evaluate the overall category performance using a quality metric
4. update the best parameter values if a better performance is found
5. choose the next set of parameter values
6. go to step 2 unless a termination criterion is reached

For step 5, there are a variety of ways to choose the next set of parameters. In our experiments, we use a generate-and-test scheme which basically selects parameter values within a specified range in a predefined manner. More advanced schemes can be adopted such as the hill-climbing scheme and the best-first search scheme. In step 3, we need a quality metric to measure the categorization performance. Several quality metrics are proposed, namely, the Category Perspective Metric, the Document Perspective Metric, and the Decision Perspective Metric. These metrics will be described in the next section.

2.4 The Category Extraction Process

This process is responsible for extracting the categories for a newly arrived document. Usually we need to accomplish it in an on-line fashion. As illustrated in Figure 1, we make use of the category

learning model to learn the desired categories. The parameters in the learning model should have been previously computed in the parameter selection process.

We evaluate the categorization performance using a new set of documents as the test collection which is different from the training collection. For our test collections, the correct (manually assigned) categories are also known for each of these documents. Therefore, we can measure the categorization performance by comparing the categories learned with the known categories using a quality metric. Similar to the parameter selection process, a variety of quality metrics can be used. However, it is essential that the metric in the parameter selection process stays consistent with the metric used in the evaluation process. We maintain this consistency within each experimental run.

3 Categorization Quality Metrics

3.1 Category Perspective Metric

This evaluation metric operates with the category as the focal point. For each category, the categorization goal is viewed as a binary classification problem. Given a category, the algorithm decides whether each document is in or not in this category. With a single category as the focus, let

a = the number of documents assigned to the category both manually and automatically.

b = the number of documents assigned to the category automatically but not manually.

c = the number of documents assigned to the category manually but not automatically.

Then two common measures, namely recall (R) and precision (P) can be defined as:

$$R = a/(a+c); \;\; P = a/(a+b)$$

We use the F-measure, a weighted combination of recall and precision proposed in [9], as the basis of our quality metric:

$$\begin{aligned} F_\beta &= [(\beta^2+1)PR]/(\beta^2 P + R) \\ &= [(\beta^2+1)a]/[(\beta^2+1)a + b + \beta^2 c] \end{aligned}$$

A common usage of this measure is to set β to 1. Hence,

$$F_1 = 2a/(2a + b + c) \tag{2}$$

The F_1 score is computed for each category in the domain and these scores are averaged to determine the mean F_1 score. Since this score averages performance across categories, we refer to this metric as the Category Perspective Metric.

3.2 Document Perspective Metric

This evaluation approach has been adopted by Yang [19]. Here the categorization results are assessed with the document as the focal point. Since categories are associated with each document with a certain strength (see Equation 1), all categories may be ranked in the context of each document. The greater the ability to rank manually assigned categories higher than others, the better is the categorization technique. A summary measure assessing this ability is the 11-AvgP (11-point average precision) score or 10-AvgP (10-point average precision) score [14]. Both scores assess the ranking of categories for each document and then take the average across documents. In our experiments, we compute both 10-AvgP and 11-AvgP scores.

3.3 Decision Perspective Metric

This evaluation scheme derives from the early work of Lewis [8]. Given a document and a category, a categorization decision is made to determine whether or not to assign this category to the document. When automatic categorization is conducted, a number of these decisions are made. Out of these decisions, some may match with the manual decisions, while others may not. This metric compares the automated decisions with the manual ones. An "assignment" is defined as the positive decision to assign a category to a document. Let

p = the number of correct assignments made automatically

q = the number of assignments made automatically

r = the number of assignments made manually.

Then, we can define "micro recall", "micro precision", and "micro F_β measure" as follows:

$$\begin{aligned} \text{micro recall} &= p/r \\ \text{micro precision} &= p/q \\ \text{micro } F_\beta \text{ measure} &= \frac{(\beta^2+1)(\text{micro precision})(\text{micro recall})}{\beta^2(\text{micro recall}) + (\text{micro precision})} \end{aligned}$$

The current literature does not yet indicate which of these three metric perspectives is the most appropriate for text retrieval. Thus we use all three perspectives with the expectation that our retrieval experiments will offer some insights on these options.

4 Experimental Results on Categorization

4.1 The HERSH Corpus

We conducted a series of experiments using 2,344 medical documents from the MEDLINE database referred to as the HERSH corpus [5]. Each document includes a free–text portion and a set of manually assigned MeSH (Medical Subject Headings) categories. In our experiments, individual words are extracted from the MeSH phrases and then stemmed. Henceforth we refer to these stemmed words as our "categories". This approach is justified since our retrieval strategy operates at the level of word stems. We conducted automatic categorization on this collection and evaluated the performance of the categorization process.

We randomly divided the HERSH corpus into two partitions, namely the training collection, referred to as TR, of 586 documents and the test collection, referred to as TE, of 1,758 documents. The division of this corpus is the same as the experiments done by some previous work such as [19, 20] so that performance comparison can be made. To make use of the training collection TR in the parameter selection process, we further divided it randomly into two sets. The first set containing 146 documents is the set of exemplars (Υ) for training. The other set containing 440 documents forms the tuning set (Θ). We make the size of Θ be three times of the size of Υ since the size of TE is three times of the size of TR. After the parameter selection process, the whole training set of 586 documents was used as the set of exemplars for categorization. To evaluate the categorization performance, we used the test

collection TE. We conducted some experiments under each quality metric mentioned above and the results are presented below.

4.1.1 Category Perspective Results on the HERSH Corpus

Three experimental runs labeled C0, C35, and C50 were conducted. They differ in the pool of categories involved. The C0 run involves all manually assigned categories which exist in both the set of training collection TR and the test collection TE. The C35 and C50 runs limit the category pool in C0 to those which occur in TR with a document frequency greater than 35 and 50 respectively. Document frequency is the number of documents to which a specific category is assigned.

Tables 1, 2 and 3 present the mean F_1 scores obtained for all different parameter combinations for the C0, C35 and C50 runs respectively in the parameter selection process. The parameter M ranges from 10 through 60 in steps of 10 while the parameter N ranges from 5 through 30 in steps of 5. The tables indicate that the desirable values for N and M are 5 and 50 respectively for the C0 run, 5 and 40 respectively for the C35 run, 30 and 20 respectively for the C50 run. These parameter values were used in the final categorization process. Table 4 summarizes the results achieved in both the parameter selection and categorization processes for all three experimental runs. The size of the category pool diminishes drastically from C0 to C35 and above. It can be seen from this table that as the frequency threshold on the category set increases, the F_1 score improves.

		M (# categories)					
		10	20	30	40	50	60
	5	0.143	0.203	0.224	0.243	0.258	0.253
	10	0.091	0.168	0.214	0.229	0.228	0.237
N	15	0.074	0.133	0.181	0.212	0.232	0.225
(#	20	0.059	0.105	0.149	0.185	0.198	0.225
docs)	25	0.053	0.090	0.138	0.166	0.186	0.195
	30	0.049	0.083	0.119	0.151	0.171	0.173

Table 1: Parameter Selection: Mean F_1 Scores for the HERSH Corpus (C0 run).

		M (# categories)					
		10	20	30	40	50	60
	5	0.348	0.454	0.464	0.468	0.462	0.450
	10	0.353	0.439	0.450	0.446	0.436	0.435
N	15	0.362	0.455	0.452	0.442	0.431	0.412
(#	20	0.337	0.444	0.451	0.432	0.420	0.409
docs)	25	0.332	0.453	0.447	0.433	0.416	0.397
	30	0.343	0.451	0.447	0.424	0.404	0.391

Table 2: Parameter Selection: Mean F_1 Scores for the HERSH Corpus (C35 run).

		M (# categories)					
		10	20	30	40	50	60
	5	0.380	0.491	0.499	0.503	0.500	0.497
	10	0.394	0.482	0.502	0.495	0.480	0.480
N	15	0.397	0.501	0.502	0.490	0.477	0.460
(#	20	0.377	0.508	0.499	0.479	0.464	0.452
docs)	25	0.374	0.508	0.495	0.476	0.461	0.445
	30	0.392	0.509	0.494	0.464	0.449	0.439
	35	0.368	0.508	0.492	0.458	0.441	0.429

Table 3: Parameter Selection: Mean F_1 Scores for the HERSH Corpus (C50 run)

	Parameter Selection Based on the TR Collection				Categorization Evaluation Based on the TE Collection	
Run	# categories	F_1 score	N	M	# categories	F_1 score
C0	641	0.258	5	50	1,461	0.19
C35	58	0.468	5	40	58	0.516
C50	43	0.509	30	20	43	0.54

Table 4: Summary of Runs Based on Category Perspective, HERSH Corpus

4.1.2 Document Perspective Results on the HERSH Corpus

In this perspective all categories are ranked in the context of a document, thus the parameter M has no relevance. Table 5 presents the parameter selection process based on the Document Perspective Metric. The ALL run represents the experiment concerning all categories appearing in either the training or the testing document collection. The TRN run represents the experiment concerning those categories appearing in the training document collection. In both experiments, the optimal parameter value for N was 15 after the parameter selection process. Table 6 summarizes the runs based on the Document Perspective Metric. For the ALL run, the 10-AvgP and the 11-AvgP for the testing collection are 0.4326 and 0.4789 respectively.

	N (# documents)					
	5	10	15	20	25	30
ALL 10-AvgP	0.3837	0.4181	0.4282	0.4167	0.4090	0.4012
ALL 11-AvgP	0.4236	0.4618	0.4771	0.4630	0.4560	0.4489
TRN 10-AvgP	0.4487	0.4774	0.4899	0.4787	0.4712	0.4643
TRN 11-AvgP	0.5064	0.5284	0.5331	0.5221	0.5153	0.5090

Table 5: Parameter Selection: Document Perspective Scores for the HERSH Corpus (D run)

	Parameter Selection Based on the TR Collection				Categorization Evaluation Based on the TE Collection		
Run	# cat.	10–AvgP	11–AvgP	N	# cat.	10–AvgP	11–AvgP
ALL	641	0.4282	0.4771	15	1,461	0.4326	0.4789
TRN	641	0.4899	0.5331	15	641	0.4921	0.5330

Table 6: Summary of Runs Based on Document Perspective, HERSH Corpus

		M (# categories)					
		10	20	30	40	50	60
	5	0.378	0.456	0.432	0.389	0.358	0.338
	10	0.388	0.470	0.475	0.446	0.407	0.371
N	15	0.394	0.474	0.476	0.454	0.421	0.385
(#	20	0.384	0.463	0.470	0.451	0.420	0.387
docs)	25	0.379	0.460	0.466	0.446	0.419	0.386
	30	0.375	0.456	0.459	0.440	0.413	0.382

Table 7: Parameter Selection: Mean Micro F_1 Scores for the HERSH Corpus (L0 run)

		M (# categories)					
		10	20	30	40	50	60
	5	0.469	0.567	0.578	0.583	0.578	0.567
	10	0.525	0.576	0.575	0.558	0.533	0.530
N	15	0.548	0.589	0.566	0.544	0.525	0.500
(#	20	0.547	0.586	0.562	0.530	0.506	0.491
docs)	25	0.546	0.588	0.556	0.523	0.497	0.471
	30	0.548	0.584	0.553	0.512	0.482	0.464

Table 8: Parameter Selection: Mean Micro F_1 Scores for the HERSH Corpus (L35 run)

		M (# categories)					
		10	20	30	40	50	60
	5	0.478	0.581	0.592	0.596	0.593	0.588
	10	0.537	0.587	0.592	0.579	0.556	0.553
N	15	0.560	0.601	0.582	0.563	0.548	0.528
(#	20	0.561	0.599	0.575	0.551	0.529	0.516
docs)	25	0.561	0.599	0.569	0.541	0.522	0.503
	30	0.565	0.596	0.565	0.530	0.508	0.495

Table 9: Parameter Selection: Mean Micro F_1 Scores for the HERSH Corpus (L50 run)

	Parameter Selection Based on the TR Collection						Categorization Evaluation Based on the TE Collection			
Run	# cat.	recall	precision	F_1	N	M	# cat.	recall	precision	F_1
L0	641	0.535	0.429	0.476	15	30	1,461	0.535	0.442	0.484
L35	58	0.694	0.512	0.589	15	20	58	0.681	0.522	0.591
L50	43	0.723	0.514	0.601	15	20	43	0.715	0.520	0.602

Table 10: Summary of Runs Based on Decision Perspective, HERSH Corpus

4.1.3 Decision Perspective Results on the HERSH Corpus

Similar to the Category Perspective Metric, three different experimental runs, L0, L35 and L50 were conducted based on the same pool of categories as used in the C0, C35 and C50 runs respectively. Tables 7, 8 and 9 show the mean micro F_1 scores achieved for L0, L35 and L50 runs in the parameter selection process. From the tables, the optimal values for N and M are 15 and 30 respectively for the L0 run, 15 and 20 respectively for the L35 run, 15 and 20 respectively for the L50 run. Table 10 gives the summary of the parameter selection and categorization evaluation runs based on the Decision Perspective Metric. The table includes micro–recall and micro–precision scores. Once again it is clear that the scores improve as the frequency threshold increases.

4.2 The OHSUMED Corpus

We conducted a series of experiments using a much larger document test corpus known as OHSUMED [6]. It is a subset of the MEDLINE database and consists of medical documents from 1987 to 1991. These documents are also manually assigned MeSH categories. In our experiment, we used those documents that have both the abstract and MeSH categories assigned. The number of documents in each year is 36,890 for 1987, 47,055 for 1988, 49,805 for 1989, 49,481 for 1990, and 50,216 for 1991. Thus the total number of documents in the OHSUMED corpus is 233,447.

The OHSUMED corpus was divided into a training collection and a test collection chronologically. We used 183,231 documents from 1987 to 1990 as the training collection and it is also used for the parameter selection process. The documents in 1991 was used as the test collection. Of the 183,231 documents in the training collection, we further divided it into two sets for the parameter selection process. The first set which consists of 133,750 documents from 1987 to 1989 was used as the set of exemplars (Υ) for training. The other set which consists of 49,481 documents from 1990 was used as the tuning set (Θ).

4.2.1 Experimental Results on OHSUMED

The experiment for the OHSUMED corpus was conducted using the Category Perspective Metric. We limited the category pool to those which occur in the corresponding exemplar set with a frequency

greater than 75. Table 11 presents the mean F_1 scores obtained for all different parameter combinations tested in the parameter selection process. It indicates that the desirable values for N and M are 20 and 30 respectively. These parameter values were used in the categorization evaluation process. Table 12 summarizes the results achieved in this experiment. It shows that the mean F_1 score for the categorization process is 0.441.

		M (# categories)						
N		10	15	20	25	30	35	40
(#	10	0.320	0.387	0.418	0.420	0.408	0.386	0.357
docs)	15	0.303	0.378	0.419	0.432	0.426	0.410	0.388
	20	0.331	0.391	0.418	0.432	0.435	0.379	0.355

Table 11: Parameter Selection: Mean F_1 Scores for the OHSUMED Corpus, Category Perspective (C75)

	Parameter Selection Based on the Training Set				Evaluation Based on the Testing Set	
Run #	# categories	F_1 score	N	M	# categories	F_1 score
OHSUMED	2725	0.435	20	30	2725	0.441

Table 12: Summary of Runs for the OHSUMED Corpus

4.3 Comparative Performance

Yang did a similar experiment for the HERSH corpus using the same training and testing collections [20]. The 10-AvgP obtained by Yang based on document perspective was 0.349. Compared with this result, our performance is quite encouraging since the 10-AvgP of our approach is 0.4326. However a difference in our studies is that Yang uses the complete MeSH phrase as a single category. In contrast our categories are the single word stems generated from the MeSH phrases.

For the OHSUMED corpus, Lewis *et. al.* conducted an experiment using the same training and testing collections on categories associated with the Heart Disease [9]. They obtained an F_1 score of 0.53 based on category perspective using the Widrow–Hoff algorithm. Yang also conducted an experiment

on the same corpus and partition using LLSF technique [21]. The F_1 score obtained was 0.55. However, both Lewis and Yang used only 119 categories associated with the Heart Disease in their experiments while we used the whole set of 2,725 categories in our experiment. Comparisons are difficult since we work with the complete OHSUMED collection. Moreover they used phrases as categories while we adopt a word stem approach, since our focus is on retrieval based on word stems.

Part II

5 Categorization For Text Retrieval

The automatic categorization method described in Part I can support a variety of applications such as text classification [7], text filtering [12], and text retrieval [16]. In the second part of the paper, we investigate the application of automatic categorization to text retrieval. Text retrieval aims at retrieving relevant documents from a document corpus given a query expressing the information need of a user. We compare text retrieval performance using documents that are automatically categorized with performance using manually categorized documents. We also assess retrieval performance against a baseline which has no categories in its documents.

5.1 Document and Query Representations

Similar to categorization, documents and queries in text retrieval are represented by vectors. However, the representation and manipulation of vectors are different from the ones used in categorization.

Each document D is represented by two vectors, namely the free–text vector and the category vector. The free–text vector is derived from the free–text portion of the document (e.g., the title and the abstract). The category vector is derived from the categories assigned to the document. In essence, a document is represented as follows:

$$D_i = (d_{i1}, d_{i2}, ..., d_{ip});\ (c_{i1}, c_{i2}, ..., c_{iq}) \tag{3}$$

where d_{ij} represents the weight of term j in D_i and p is the vocabulary size of the free–text portions of all documents in the corpus. Similarly, c_{ij} represents the weight of category j in document D_i and q is the vocabulary size of all categories.

We choose this representation scheme for retrieval purposes based upon our previous work [16]. However, this technique has previously assumed the manual assignment of categories to each document. Instead we now apply the automatic categorization technique described in Part I so that the manual assignment step can be eliminated. This is particularly useful when human experts on categorization are not available or not affordable.

Each query, similar to a document, is represented by a free–text vector and a category vector. The free–text vector is constructed by the same method used for the free–text vectors of the documents. Since the natural language queries do not arrive with search criteria identified, we use two different ways to design the category vectors for queries:

- Simple-Query design : The query's free–text vector is copied to form the query's category vector.
- Advanced-Query design: The query's category vector is constructed by applying the category learning model described in Part I. More specifically, the free–text vector is used to conduct an initial retrieval run on the corpus. The top U documents retrieved are analyzed in terms of their categories. From these, the top V categories are extracted to form the category vector. This strategy was explored successfully for the MEDLINE collection in our previous work [16].

5.2 The Retrieval Model

The retrieval step is conducted by computing the similarity between a document D and a query Q as follows:

$$\Delta(D, Q) = \mu\Delta(D_{\text{free-text}}, Q_{\text{free-text}}) + \Delta(D_{\text{category}}, Q_{\text{category}}) \tag{4}$$

where $D_{\text{free-text}}$ and D_{category} are the free–text vector and the category vector for D respectively. Similarly, $Q_{\text{free-text}}$ and Q_{category} are the free–text vector and the category vector for Q respectively.

μ is a parameter that allows one to control the relative emphasis on the two types of vectors during retrieval. This technique allows the retrieved documents to be ranked against the query.

In each retrieval experiment, the documents are ranked with respect to each query. To evaluate the retrieval performance, we compute the average of precision scores of all queries at 11 recall points starting from 0.0 to 1.0 in steps of 0.1. Then the average of these 11 precision scores is computed to get a single measure. This measure becomes the 11–point average precision (11–AvgP) score for evaluating the retrieval performance. This averaging technique yields *macro average* data wherein each query is allowed to contribute equally to the overall performance score for the system [14].

6 Experimental Design

The HERSH corpus in the categorization experiment in Part I was also used in our text retrieval experiment. The automatic categorization strategies that yielded the best performance in Part I form the basis of our retrieval experiments. The retrieval experiment was conducted on the test collection subset (TE) composed of the 1,758 documents from the HERSH corpus. In Part II here, we refer to this collection as the RTT (ReTrieval Test) collection. The HERSH corpus is accompanied by a set of queries which are in the form of simple natural language expressing an information need. For each query, there is a set of relevant documents which have been manually judged. Thus the retrieval performance can be measured by comparing the documents retrieved by the system and the ones manually judged. We chose those queries that have at least one relevant document in the RTT collection. There are 73 queries satisfying this requirement.

The best strategies within each of the three evaluation perspectives: Category, Decision and Document were tested for retrieval performance. Each strategy is assessed against two baselines [2]:

- Baseline 1 (B1): Retrieval without MeSH categories (*i.e.*, retrieval using free–text alone).
- Baseline 2 (B2): Retrieval using free text and manually assigned MeSH categories.

To conduct text retrieval experiments, we make use of the SMART system [14] since it supports the

[2]Note that the stems of the individual words of the MeSH phrases form our domain of categories for these experiments.

vector space model.

6.1 Document Representations

SMART allows a wide variety of weighting schemes for computing term weights. Each scheme is represented by a triple: $\mathcal{ABC}$. $\mathcal{A}$ represents the term frequency component, *i.e.*, the number of times the term occurs in the document. $\mathcal{B}$ represents the inverse document frequency component which increases with the rarity of the term in the database. $\mathcal{C}$ represents the normalization component for the length of the document. Based on results from our prior experimentation with the same test corpus [16, 18], we used the **atn** schemes for documents: **a** stands for augmented term frequency, **t** represents the inverse document frequency factor and **n** represents no normalization for length. If we describe this scheme more precisely, the weight of a term in a document is given by:

$$(0.5 + 0.5 * tf/m) * \log(R/n). \tag{5}$$

where R is the number of documents in the RTT collection; n is the number of documents which contain the term being considered; tf is the frequency of the term in the document; and m is the maximum tf value for the current document. The objective of division by m is to normalize the term frequency by the maximum tf observed in the document. The term $\log(R/n)$ corresponds to the inverse document frequency weight.

A total of 9 different document representations were tested. All representations include a free–text vector. The difference is in their MeSH category vectors.

- Representation 1 (Baseline 1): No MeSH category vector.
- Representation 2 (Baseline 2): The MeSH category vector is formed from manual categorization.
- Representations 3-5: The MeSH category vector is derived by automatic categorization based on the Category Perspective Metric as described in Section 4.1.1. The three best strategies, one each from C0, C35 and C50 were tested.
- Representations 6-8: The MeSH category vector is derived by automatic categorization based on the Decision Perspective Metric as described in Section 4.1.3. The three best strategies, one each

from L0, L35 and L50 were tested.

- Representation 9: The MeSH category vector is derived by automatic categorization based on the Document Perspective Metric as described in Section 4.1.2.

6.2 Query Representations

Each of the 73 test queries arrives as a simple natural language expression of an information need. Some queries are expressed in full sentences, others are incomplete. In each case, a free–text vector was derived analogous to the document representation in Equation 3. Based on our previous work [16], term weights were determined using the **atc** scheme in SMART. This is similar to **atn** used in document representation due to **at** being in common. The difference is that term weights, due to the **c** factor, are normalized by the following factor:

$$\sqrt{{t_1}^2 + \ldots + {t_p}^2}$$

where t_i is the weight of the term i in the query and p is the size of the free–text vector.

6.3 Retrieval Strategies

Given that a retrieval strategy may be defined as a combination of a document representation strategy and a query representation strategy, a total of 17 retrieval strategies were tested. Each of the 8 document representations involving both the free–text and the MeSH category vectors may be combined with both the Simple-Query design and the Advanced-Query design. The Baseline 1 strategy involves only free–text. The two parameters U and V used in the Advanced-Query design were each varied independently across 5, 10, 15, and 20. The parameter μ was varied across 0.75, 1.0, 1.25, 1.5, 1.75, 2.0. Thus a total of 96 different parameter combinations were tested in experimentation involving advanced queries. For the Simple-Query design, the only parameter involved is μ and thus 6 parameter combinations were tested.

7 Retrieval Results

Table 13 shows the retrieval results measured by the 11–AvgP score for the 17 different retrieval strategies tested. The table includes the scores for the Baseline 1 (B1) and Baseline 2 (B2). The Baseline 2 scores reported are different for the Simple-Query design and the Advanced-Query design options. Note that although these two design options have identical document representations, they differ in their query representations. The remaining rows represent performance for different automatic categorization strategies. For example, row 11 reports the results when the best document categorization strategy in Part I under the C0 framework is combined with the Advanced-Query design. This yields a 11–AvgP score of 0.5619 which offers a significant (9.3%) improvement over the Baseline 1 performance (0.5143) while being significantly worse (−7.9%) than the Baseline 2 performance of 0.6098. This row also indicates that the query was created using the query's initial free–text vector to retrieve the top 20 documents. Then, the MeSH categories assigned to these 20 documents were analyzed and the best 20 were used to create the query's MeSH category vector. It also indicates that the parameter μ in Equation 4 was set to 1.25 during retrieval.

It can be seen from Table 13 that the manual strategies (rows 2 and 10) are significantly superior to the Baseline 1 strategy. This is consistent with previous results suggesting that MeSH should not be ignored during retrieval [16]. Similarly, there is a 9.5% improvement between manual strategies when one moves from simple to advanced queries. This result is also consistent with previous studies which have shown that investing in retrieval feedback for query design yields good returns [16].

As regards the automatic strategies, we continue to see improvements when moving from the Simple-Query design to the Advanced-Query design. The best Simple-Query performance is 0.5106 and that for Advanced-Query is 0.5619 (10.0% improvement). The table indicates that automatic categorization performs worse than manual categorization. Further analysis of the underlying vocabularies yields the explanation for this. Specifically since automatic categorization is done based on a set of exemplar documents, the categorization vocabularies (free–text and MeSH) for the automatic collection are limited to the vocabularies of the training collection. However for the manual runs, the vocabularies come from the original vocabulary set of much larger size. Thus, for example, the MeSH category vectors

Row	MeSH Approach	11–AvgP	% diff. wrt. B1	% diff. wrt. B2	U	V	μ
1	B1: No MeSH	0.5143	na	na	na	na	na
Simple-Query design							
2	B2: Manual MeSH	0.5632	9.5%*	na	na	na	1.25
3	C0	0.5106	-0.7%	-9.3%*	na	na	1.0
4	C35	0.5103	-0.8%	-9.4%*	na	na	1.0
5	C50	0.5038	-2.0%	-10.5%*	na	na	1.5
6	L0	0.5083	-1.2%	-9.7%*	na	na	2.0
7	L35	0.5045	-1.9%	-10.4%*	na	na	1.5
8	L50	0.5045	-1.9%	-10.4%*	na	na	1.5
9	D	0.5080	-1.2%	-9.8%*	na	na	1.75
Advanced-Query design							
10	B2: Manual MeSH	0.6098	18.5%*	na	10	15	1.25
11	C0	0.5619	9.3%*	-7.9%*	20	20	1.25
12	C35	0.5533	7.6%*	-9.2%*	10	20	1.5
13	C50	0.5331	3.7%	-12.5%*	20	20	1.75
14	L0	0.5526	7.4%*	-9.4%*	20	20	1.25
15	L35	0.5449	5.9%*	-10.6%*	20	20	1.25
16	L50	0.5449	5.9%*	-10.6%*	20	20	1.25
17	D	0.5524	7.4%*	-9.4%*	5	10	1.0

Table 13: Retrieval Performance (Vocabulary Differences Not Controlled). Asterisk Denotes the Difference is Significant ($p < 0.01$) Using the Non–Parametric Wilcoxon Signed Rank Test for Matched Samples. "na" Denotes "not applicable"

assigned manually (rows 2 and 10) for the RTT collection were generated using a MeSH vocabulary of 2,968 word stems. However the MeSH category vectors generated automatically for the same collection were produced from the 586 documents in the training collection of the HERSH corpus from Part I. This MeSH vocabulary base contains only 1,604 word stems. This difference in underlying vocabularies may explain the difference in performance. In order to conduct a more meaningful comparison, we repeated our retrieval experiment involving the 17 retrieval strategies by controlling for this vocabulary difference. In other words the MeSH categories not existing in the training collection were removed from the manually generated representations for RTT documents.

Table 14 presents the result for this second set of retrieval runs. This table indicates that when vocabulary differences are controlled, all automatic retrieval runs are better than retrieval without MeSH [3]. Both simple and advanced queries show better performance than the results without controlling the

[3]Interestingly, manual MeSH combined with simple queries (row 2) does not yield improved results compared with

Row	MeSH Approach	11–AvgP	% diff. wrt. B1	% diff. wrt. B2	U	V	μ
1	B1: No MeSH	0.5050	na	na	na	na	na
Simple–Query design							
2	B2: Manual MeSH	0.5078	0.6%	na	na	na	1.75
3	C0	0.5438	7.7%*	7.1%*	na	na	1.0
4	C35	0.5448	7.9%*	7.3%*	na	na	1.0
5	C50	0.5332	5.6%*	5.0%*	na	na	1.5
6	L0	0.5402	7.0%*	6.4%*	na	na	2.0
7	L35	0.5362	6.2%*	5.6%*	na	na	2.0
8	L50	0.5362	6.2%*	5.6%*	na	na	1.0
9	D	0.5456	8.0%*	7.4%*	na	na	1.75
Advanced–Query design							
10	B2: Manual MeSH	0.5677	12.4%*	na	10	20	1.0
11	C0	0.5757	14.0%*	1.4%	10	20	1.25
12	C35	0.5754	13.9%*	1.4%	15	15	1.0
13	C50	0.5435	7.6%*	-4.3%	20	20	1.75
14	L0	0.5582	10.5%*	-1.7%	10	20	1.5
15	L35	0.5506	9.0%*	-3.0%	10	20	2.0
16	L50	0.5506	9.0%*	-3.0%	20	20	2.0
17	D	0.5665	12.2%*	-0.2%	5	15	1.25

Table 14: Retrieval Performance (Vocabulary Differences Controlled). Asterisk Denotes the Difference is Significant ($p < 0.01$) Using the Non–Parametric Wilcoxon Signed Rank Test for Matched Samples. "na" Denotes "not applicable"

vocabulary differences.

7.1 Effect of Parameter Values on Retrieval Performance

The results in Tables 13 and 14 reveal only a piece of the picture regarding retrieval. In particular each row presents only the best performance achieved over all the parameter combinations (U, V, and μ where relevant) tested for that row. In the Advanced-Query design, U is the number of top ranking documents examined, V is the number of MeSH categories added to the query and μ participates in all retrieval processes where both free–text and MeSH are involved. For example the 0.5448 score of row 4 in Table 14 represents the best result over 6 parameter μ values. Similarly the 0.5754 score of row 12 represents the best from 96 parameter combinations tested under the "C50 Advanced Query framework". To get a more complete picture we now present a query–by–query analysis that explores the effect of parameter combinations on retrieval performance.

We explain our analysis strategy with reference to Table 15. This table provides the results of a query–by–query statistical analysis of the 96 parameter combinations tested for C0 combined with the Advanced–Query design. It compares retrieval performance achieved using the automatic strategy with the retrieval performance achieved using the baselines B1 and B2. "+" indicates that the automatic strategy is significantly better than the corresponding manual strategy statistically. "-" indicates that the manual strategy is significantly better than the automatic strategy statistically. "=" indicates no significant difference between the two statistically. The statistical analysis is based on a query–by–query analysis using the Wilcoxon signed rank test. Straightforward counts indicate that C0 is always significantly better than the B1. Moreover it is better than the B2 in 43 cases, worse in 44 cases and equivalent in the remaining 9 cases. Thus it can be observed that within the "C0 Advanced-Query design" framework, the automatic strategy always performs better than the baseline retrieval with no MeSH involved. Also, across all 96 parameter combinations tried, automatic categorization is equivalent to the Baseline 2 manual categorization.

Table 16 provides the query–by–query analysis for C35. This table shows that C35 also possesses a

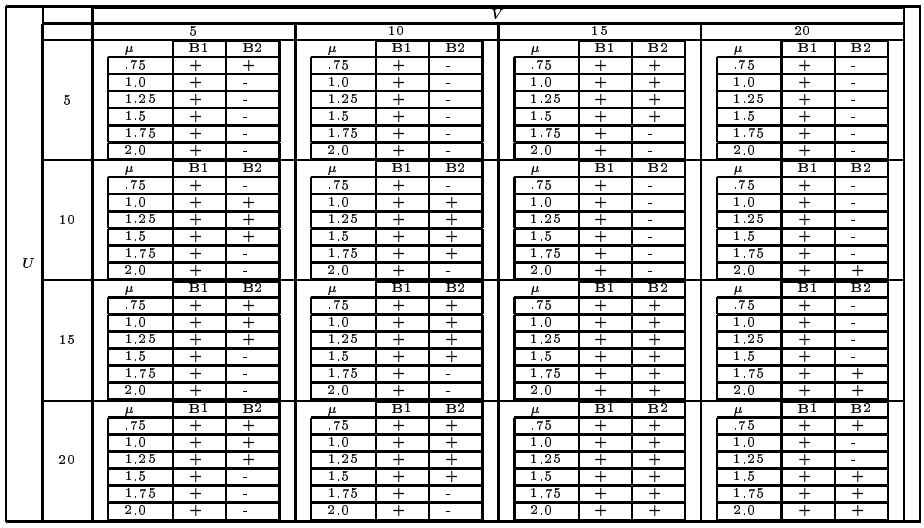
no MeSH (row 1). This result is different from the results from our previous study [16] and is also explained by the vocabulary differences.

U	V = 5			V = 10			V = 15			V = 20		
	μ	B1	B2	μ	B1	B2	μ	B1	B2	μ	B1	B2
5	.75	+	+	.75	+	=	.75	+	=	.75	+	-
	1.0	+	+	1.0	+	=	1.0	+	+	1.0	+	-
	1.25	+	-	1.25	+	-	1.25	+	-	1.25	+	-
	1.5	+	-	1.5	+	=	1.5	+	-	1.5	+	-
	1.75	+	-	1.75	+	-	1.75	+	-	1.75	+	-
	2.0	+	-	2.0	+	-	2.0	+	-	2.0	+	-
	μ	B1	B2	μ	B1	B2	μ	B1	B2	μ	B1	B2
10	.75	+	+	.75	+	=	.75	+	-	.75	+	-
	1.0	+	+	1.0	+	+	1.0	+	-	1.0	+	-
	1.25	+	+	1.25	+	+	1.25	+	-	1.25	+	-
	1.5	+	+	1.5	+	+	1.5	+	-	1.5	+	-
	1.75	+	+	1.75	+	+	1.75	+	-	1.75	+	-
	2.0	+	+	2.0	+	-	2.0	+	-	2.0	+	=
	μ	B1	B2	μ	B1	B2	μ	B1	B2	μ	B1	B2
15	.75	+	+	.75	+	=	.75	+	+	.75	+	-
	1.0	+	+	1.0	+	+	1.0	+	+	1.0	+	-
	1.25	+	+	1.25	+	+	1.25	+	+	1.25	+	-
	1.5	+	-	1.5	+	+	1.5	+	+	1.5	+	-
	1.75	+	-	1.75	+	+	1.75	+	+	1.75	+	-
	2.0	+	-	2.0	+	-	2.0	+	=	2.0	+	-
	μ	B1	B2	μ	B1	B2	μ	B1	B2	μ	B1	B2
20	.75	+	=	.75	+	+	.75	+	+	.75	+	+
	1.0	+	+	1.0	+	+	1.0	+	+	1.0	+	+
	1.25	+	-	1.25	+	+	1.25	+	+	1.25	+	+
	1.5	+	-	1.5	+	+	1.5	+	+	1.5	+	+
	1.75	+	-	1.75	+	+	1.75	+	+	1.75	+	+
	2.0	+	-	2.0	+	-	2.0	+	+	2.0	+	+

Table 15: Statistical Analysis of C0 (Vocabulary Differences Controlled) + Implies the Automatic Method is Significantly Better Statistically. - Implies the Automatic Method is Significantly Worse Statistically. = Implies that they are Statistically Similar.

similar balance by being better in 45 cases of the parameter settings and worse in 51 cases. We perform the query–by–query analysis only for the situation where the vocabularies are controlled. Table 17 shows a summary of the comparison between B2 and different automatic categorization strategies tested for retrieval. It should also be noted that the automatic strategies are almost always significantly better than B1 statistically.

Table 17 provides some interesting results in that it allows us to differentiate between perspectives. It is clear that C0 and C35 are distinct from the remaining perspectives. In fact the Decision and Document perspectives yielded poor results based on a query–by–query analysis. If we consider that behind each query lies a distinct user, this study recommends the use of the Category Perspective Metric over the other metrics for text retrieval for the HERSH corpus.



U	V = 5			V = 10			V = 15			V = 20		
	μ	B1	B2	μ	B1	B2	μ	B1	B2	μ	B1	B2
5	.75	+	+	.75	+	-	.75	+	+	.75	+	-
	1.0	+	-	1.0	+	-	1.0	+	+	1.0	+	-
	1.25	+	-	1.25	+	-	1.25	+	+	1.25	+	-
	1.5	+	-	1.5	+	-	1.5	+	+	1.5	+	-
	1.75	+	-	1.75	+	-	1.75	+	-	1.75	+	-
	2.0	+	-	2.0	+	-	2.0	+	-	2.0	+	-
10	μ	B1	B2	μ	B1	B2	μ	B1	B2	μ	B1	B2
	.75	+	-	.75	+	-	.75	+	-	.75	+	-
	1.0	+	+	1.0	+	+	1.0	+	-	1.0	+	-
	1.25	+	+	1.25	+	+	1.25	+	-	1.25	+	-
	1.5	+	+	1.5	+	+	1.5	+	-	1.5	+	-
	1.75	+	-	1.75	+	+	1.75	+	-	1.75	+	-
	2.0	+	-	2.0	+	-	2.0	+	-	2.0	+	+
15	μ	B1	B2	μ	B1	B2	μ	B1	B2	μ	B1	B2
	.75	+	+	.75	+	+	.75	+	+	.75	+	-
	1.0	+	+	1.0	+	+	1.0	+	+	1.0	+	-
	1.25	+	+	1.25	+	+	1.25	+	+	1.25	+	-
	1.5	+	-	1.5	+	+	1.5	+	+	1.5	+	-
	1.75	+	-	1.75	+	-	1.75	+	+	1.75	+	+
	2.0	+	-	2.0	+	-	2.0	+	+	2.0	+	+
20	μ	B1	B2	μ	B1	B2	μ	B1	B2	μ	B1	B2
	.75	+	+	.75	+	+	.75	+	+	.75	+	+
	1.0	+	+	1.0	+	+	1.0	+	+	1.0	+	-
	1.25	+	+	1.25	+	+	1.25	+	+	1.25	+	-
	1.5	+	-	1.5	+	+	1.5	+	+	1.5	+	+
	1.75	+	-	1.75	+	-	1.75	+	+	1.75	+	+
	2.0	+	-	2.0	+	-	2.0	+	+	2.0	+	+

Table 16: Statistical Analysis of C35 with Controlled Vocabulary. + Represents that the Automatic Method is Significantly Better Statistically. - Represents that the Automatic Method is Significantly Worse Statistically. = Represents that they are Statistically Similar.

	Number of instances where strategy is		
Strategy	Better than B2	Similar to B2	Worse than B2
C0	43	9	44
C35	45	0	51
C50	0	0	96
L0	0	0	96
L35/L50	1	0	95
D0	16	2	78

Table 17: Summary of Query–by–Query Analysis Comparing B2 and Different Automatic Categorization Strategies (Vocabulary Differences Controlled).

8 Conclusion

We develop a new approach to automatic text categorization. Once the categorization model is learned, it can be used for classifying future documents. The categorization approach derives from a machine learning paradigm known as instance-based learning and an advanced document retrieval technique known as retrieval feedback. We demonstrate the effectiveness of our categorization approach using two MEDLINE test collections namely the HERSH corpus and the OHSUMED corpus.

Apart from developing an automatic categorization approach, we also investigate the application of this categorization process to advanced text retrieval. Our experiments clearly indicate that the categorization process is effective. It improves the retrieval performance compared with no categorization. It also achieves the retrieval performance equivalent to the results using manual categorization. This is concluded on the basis of analyzing the retrieval performance for each individual test query statistically. Moreover our study indicates that the Category Perspective Metric is the one most suited for text retrieval for the HERSH corpus. Finally this paper shows that automatic categorization is effective for advanced text retrieval.

References

[1] C. Apte, F. Damerau, and S. M. Weiss , "Automated Learning of Decision Rules for Text Categorization", in *ACM Transactions on Information Systems*, Vol. 12, No. 3, pp. 233–251, 1994.

[2] C. Buckley, G. Salton, J. Allan, and A. Singhal, "Automatic Query Expansion using SMART: TREC-3 Report",in *TREC-3, Proceedings of the Third Text REtrieval Conference*, pp. 69-80, 1995.

[3] W. W. Cohen and Y. Singer, "Context-sensitive Learning Methods for Text Categorization", in *Proceedings of the Nineteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 307–315, 1996.

[4] D. K. Harman, "Overview of the Third Text REtrieval Conference (TREC-3)", in *TREC-3, Proceedings of the Third Text REtrieval Conference*, pp. 1-19, 1995.

[5] W. Hersh, D. Hickam, R. Haynes, and K. McKibbon, "A Performance and Failure Analysis of SAPHIRE with a MEDLINE Test Collection". in *Journal of the American Medical Informatics Association*, Vol. 1, No. 1, pp. 51–60, 1194.

[6] W. Hersh, C. Buckley, T. Leone, and D. Hickam, "OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research", in *Proceedings of the Seventeenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 192–200, 1994.

[7] L. S. Larkey and W. B. Croft, "Combining Classifiers in Text Categorization", in *Proceedings of the Nineteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 289–297, 1996.

[8] D. D. Lewis, "Feature Selection and Feature Extraction for Text Categorization", in *Proceedings of Speech and Natural Language Workshop*, Arden House, pp. 212–217, 1992.

[9] D. D. Lewis, R. E. Schapire, J. P. Callan and R. Papka, "Training Algorithms for Linear Text Classifiers", in *Proceedings of the Nineteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 298–306, 1996.

[10] B. Masand, G. Linoff and D. Waltz, "Classifying News Stories Using Memory Based Reasoning", in *Proceedings of the Fifteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 59–65, 1992.

[11] R. Mehnert, *Federal Agency and Federal Library Reports: National Library of Medicine*, Bowker Annual: Library and book Trade Almanac, 2nd edition, pp. 110-115, 1997.

[12] J. Mostafa, S. Mukhopadhyay, W. Lam, and M. Palakal, "A Multi-Level Approach to Intelligent Information Filtering: Model, Systems, and Evaluation", to appear in *ACM Transactions on Information Systems*, 1997.

[13] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford, "Okapi at TREC-3", in *TREC-3, Proceedings of the Third Text REtrieval Conference*, pp. 109-126, 1995.

[14] G. Salton, *The Smart System – Experiments in Automatic Document Processing*, Englewood Cliffs, NJ: Prentice Hall, 1971.

[15] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley, 1989.

[16] P. Srinivasan, "Query Expansion and MEDLINE", in *Information Processing & Management*, Vol. 32, No. 4, pp. 431-443, 1996.

[17] P. Srinivasan, "Optimal Document-Indexing Vocabulary for MEDLINE", in *Information Processing and Management*, Vol. 32, No. 5, pp. 503-514, 1996.

[18] P. Srinivasan, "Retrieval Feedback in MEDLINE", in *Journal of the American Medical Informatics Association*, Vol. 3, No. 2, pp. 157-167, 1996.

[19] Y. Yang, and C. D. Chute, "An Example-Based Mapping Method for Text Categorization and Retrieval", in *ACM Transactions on Information Systems*, Vol. 12, No. 3, pp. 252–277, 1994.

[20] Y. Yang, "Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval", in *Proceedings of the Seventeenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 13–22, 1994.

[21] Y. Yang, "An Evaluation of Statistical Approaches to MEDLINE Indexing", in *Proceedings of the 1996 AMIA Annual Fall Symposium*, pp. 358-362.

Biographies

Wai Lam:

Wai Lam received a Ph.D. in Computer Science from the University of Waterloo, Canada in 1994. He worked as a visiting Research Associate at Indiana University Purdue University Indianapolis in 1995 and as a Postdoctoral Fellow at Distributed Adaptive Search Laboratory in University of Iowa in 1996. Currently he is an Assistant Professor at Department of Systems Engineering and Engineering Management in the Chinese University of Hong Kong. His current interests include data mining, intelligent information retrieval, machine learning, reasoning under uncertainty, and digital library.

Miguel Ruiz:

MIguel Ruiz is a PhD. student at the University of Iowa in the Interdisciplinary PhD. Program. He is also Assistant professor of the School of System Engineering at the University of los Andes, Venezuela. His research areas include information retrieval, machine learning methods, and parallel and distributed systems. He is member of the Information Retrieval Group of The University of Iowa.

Padmini Srinivasan:

Padmini Srinivasan is Associate professor of the School of Library and Information Science (SLIS) and Department of Management Science at The University of Iowa. She is the current Director of the SLIS. She has numerous publications in the area of Information retrieval. Her research areas include Information Retrieval, Automatic text categorization, medical informatics.