The Effect of the Quality of Pre-Assigned Subject Categories on the Text Categorization Performance

Kyung Shim*
Young-Mee Chung**

Abstract

In text categorization a certain level of correctness of labels assigned to training documents is assumed without solid knowledge on that of real-world collections. Our research attempts to explore the quality of pre-assigned subject categories in a real-world collection, and to identify the relationship between the quality of category assignment in training set and text categorization performance. Particularly, we are interested in to what extent the performance can be improved by enhancing the quality (i.e., correctness) of category assignment in training documents. A collection of 1,150 abstracts in computer science is re-classified by an expert group, and divided into 907 training documents and 227 test documents (15 duplicates are removed). The performances of before and after re-classification groups, called Initial set and Recat-1/Recat-2 sets respectively, are compared using a *k*NN classifier. The average correctness of subject categories in the Initial set is 16%, and the categorization performance with the Initial set shows 17% in F_1 value. On the other hand, the Recat-1 set scores F1 value of 61%, which is 3.6 times higher than that of the Initial set.

1. Introduction

Text categorization (TC) is an automated process to assign documents into a predefined set of categories (or labels) or vice versa. In text categorization a document can be assigned to a single category or more than one category if necessary. According to the ways of using a text classifier either a document can be classified to categories or categories may be assigned to a document which is called category-pivoted categorization and document-pivoted categorization respectively. This distinction is more pragmatic than conceptual since the former is suitable when a new

* Director. Systems R&D Center, Iris.Net (shim@irisnet.co.kr)
** Professor. Dept of Library and Information Science, Yonsei University (ymchung@yonsei.ac.kr)

category may be added to the predefined set of categories, while the latter is typically suitable for applications involving streams of incoming documents with time. As the latter situation is more common than the former, the latter is used more often (Sebastiani 2002).

Until in the late 1980s a knowledge engineering approach based on human expert knowledge was a popular one in text categorization, but from the early 1990s a machine learning approach began to gain strength due to its portability to other domains, efficiency in terms of human labor, and a reasonable effectiveness. In the machine learning paradigm an automatic text classifier is automatically built by a general inductive process. This inductive process learns the characteristics of the categories of interest. Here the object of learning is not the meaning of a text, rather a coarser-grained notion of knowledge about the text. Guthrie, Guthrie and Leistensnider (1999) called this knowledge the "flavor" of a text. All the techniques underpinning the current text categorization algorithms are basically statistical in the sense that they operate on the statistical behavior of terms in collections.

In text categorization, because texts in the collection cannot be directly processed by a classifier or by a classifier-building algorithm, documents must be transformed into a compact representation before building a classifier. This process is often called pre-processing. Pre-processing consists of word stemming and function word removal, feature selection, and term weighting. Word stemming standardizes word's suffixes and function word removal eliminates topic-neutral words such as articles. Feature selection, a.k.a., term reduction or dimensionality reduction, is performed using a function to reduce the number of dimensions by eliminating non-informative terms while maintaining meaningful ones.

So far a majority of previous experiments has concentrated on proving the superiority of proposing techniques such as indexing procedures (text representation techniques), feature reduction methods, classifier algorithms, and others. On the other hand, it is rather surprising that all the research has not paid much attention to the properties of document collections, and the relationships between the properties and categorization effectiveness. The properties of document collections include not only the quantitative characteristics such as the number of categories and documents per category, but also the qualitative aspect like the quality of category assignment in training documents.

In the machine learning approach a basic assumption is a set of training documents pre-classified by human experts is available, and subsequently a certain level of quality of the classification is tacitly taken for granted. Unfortunately, however, it is widely known that manually assigning one or more categories from a predefined set of subjects

to a document is a subjective process, and the "subject indexing is indeterminate and probabilistic beyond a certain point" (Bates 1986, p. 357). The phenomenon of indexer inconsistency that two indexers or even the same indexer at different times might assign different categories on the same document has been a disturbing and nagging problem in the field of library and information science. However, in the machine learning paradigm a reasonable quality of pre-assigned subject categories in training set is assumed, and the categorization performance is measured against a classifier that is trained on the dataset with this deemingly imperfect classification.

As aforementioned, while research on text categorization techniques is abundant, systematic approach to the properties of collections or the relationship between the properties and categorization performance is rarely found. Yet, we know neither the quality levels of classification in real world collections nor the relations between the properties and categorization effectiveness.

In our research we attempted to explore the quality of pre-assigned subject categories in a real-world collection, and to identify the relationship between the quality of category assignment in training set and text categorization performance. Particularly, we are interested in to what extent the performance can be improved by enhancing the quality (i.e., correctness) of manual category assignment in training documents.

.

2. Quality of Pre-Assigned Categories in Training Set and Machine Learning

In text categorization it is generally assumed that a set of pre-classified documents is available for training a classifier, and this classification (i.e., category assignments) is conducted manually by human indexers. It is also assumed that the subject categories assigned to the training set are correct or at least have a certain level of classification correctness. Then, it is plausible that the categorization effectiveness would be influenced by the quality of pre-assigned categories.

A majority of previous research experimented on standard test collections such as Reuters collections and OSHUMED collection is presumably based on the above assumption. However, in a text categorization research on 10 USENET newsgroups, Weiss, Kasif, and Brill (1996) reported that the accuracy of postings on 10 newsgroups averaged to 85%. Although this result might be taken to be misleading as the subject category of an article in a newsgroup is given by a person who posts it rather than an expert indexer, the problem of indexing inconsistency in manual classification or

subject indexing by human experts has been around for many years and has a long history of research (Hooper 1965; Hurwitz 1969; Jacoby, and Slamecka 1962). Before embarking the issue of indexing inconsistency and machine learning, it might be helpful to clarify our view to indexing inconsistency and classification accuracy. Indexing inconsistency (or indexing consistency) is not about accuracy, or right or wrong, rather it is about diversity in indexing decisions or "indeterminism". However, it might be unmistaken to say that indexing inconsistency is an indicator of indexing accuracy to some extent. For the purpose of the research, we hypothesize that indexing inconsistency roughly equals to indexing accuracy and the latter would be enhanced by manually re-classifying the same set of documents.

The indexing inconsistency includes inter-indexer inconsistency (different indexers tend to assign different categories to the same document) and intra-indexer inconsistency (an indexer might assign different categories to the same document at different time). Blair (1986) attributes the reason for such inconsistency, particularly the inter-indexer inconsistency, to there being no precise, specifiable procedure for deciding whether a given subject description should be assigned to a given document.

Studies of inter-indexer consistency report varying rates of inconsistency. Much of the variance, according to Chen et al. (1994), is due to (1) the nature of the subject area, (2) the levels of training and experience of the indexers involved, and (3) the controlled vocabulary nature of the index terms. As to the first one, if we take an example of botany and sociology, the language of botany is more normative than that of sociology and then it is less likely for the former to induce indexing inconsistency. And the rest of the above reasons are self-explanatory.

In a recent study using two different methods Leininger (2000) reports 49.02% and 63.32% of overall indexing consistency, and 44.17% and 45.00% of classification code consistency. Regardless of the measures adopted, indexing or classification accuracy cannot reach 100% perfect level.

Automatic text classifiers are trained on those imperfect classifications and apparently there is no way for a classifier to reach the perfect level in categorization effectiveness. In the context of text categorization, Apté, Damerau, and Weiss (1994) precisely point out the situation as follows:

> From a machine-learning perspective in this application, we know that such performance [100%] is not achievable because many labels are not correct … We observed a few topic assignment mistakes in the Reuters collection. Even with a careful reading, human language is not precise enough to allow full agreement by readers of all stories. Machine-learning programs can operate in this uncertain environment and find patterns that separate the populations with some degree of error …We can look at the recall figures as a measure of overlap or consistency with the human indexers

> of the documents. There have been studies of the consistency of human indexers with each other, although not in this context. In a survey of this work, Saracevic (1991) reported consistency values ranging from 10% to 80%. In studies comparing indexing of inadvertently duplicated documents in *Information Science Abstracts* and *MEDLINE*, consistency for central concepts or main headings, which are roughly analogous to our subject codes, was 52% to 61%... (p. 248~249)

In the conclusion of his extensive review article, Sebastiani (2002) also describes the imperfect nature of manual text categorization and automatic text categorization as well:

> It has reached effectiveness levels comparable to those of trained professionals. The effectiveness of manual TC is not 100% any way (Cleverdon 1984) and, more importantly, it is unlikely to be improved substantially by the progress of research … (p. 41)

The basic assumption of text categorization requires a set of pre-classified documents in order to train a classifier and the quality of manual indexing is far below 100%. Logically, the effectiveness of text categorization cannot exceed that of manual indexing. Despite, it is interesting that the performances of classifiers trained against a set of documents indexed by human indexer(s) are over 80% (Apté, Damerau, and Weiss 1994; Joachims 1998; Weiss et al. 1999; Yang 1999). Putting aside the variance caused by different measures, the average indexing consistency cited above largely around 50% only. How can we explain the relationship between the relatively low manual indexing consistency and the unrealistically high performance of text categorization? This phenomenon is directly contradictory to the assumption made at the beginning of this section. Does this mean that the performance of text categorization is not affected by the quality of pre-assigned categories? Or, the higher performance in TC in previous studies may be yielded because the qualities of indexing in the standard benchmark collections (Reuters collections) is higher than the indexing consistency cited above. Or, the discrepancy between lower manual indexing consistency and higher text categorization performance may be due to different criteria. That is, indexing consistency is measured by a certain criterion. On the other hand, the effectiveness of text categorization is measured by the human indexing decisions reflected on the test collection that proved to be less than perfect. The problem of measuring and measurement criteria is beyond the scope of this paper.

Concerning the basic assumption of text categorization and its relationship to categorization performance it is necessary for us to extend our knowledge. First of all, collecting the information about the accuracy of pre-assigned categories of real-world collections as well as standard benchmark collections would be a stepping stone for comparing the performance of different collections. Second, in addition to exploring the relationship between the quality of classification and categorization performance, if

there is any, to what extent the former is responsible for performance should be identified. This will guide the ways to improve categorization performance along with the development of new algorithms and techniques. With the growth of knowledge in relation to the basic assumption of TC, to a certain degree we may predict the level of performance expectancy in an experiment or even the upper bound limit of automatic text categorization.

3. Experimental Design

3.1 **Overview**

Our experiments are focused on a qualitative property of manually classified document collections: the quality of pre-assigned subject categories. It is one of our aims to examine whether there is a relationship between the quality of them and categorization performance.

In a pre-processing phase, the Porter Stemming algorithm is applied to text words and the selected word-stems are compared against the SMART stop word list to remove function words. The terms, i.e., features, are selected from the abstracts only. For feature selection terms are sorted by document frequency (DF) in descending order and only the terms satisfying DF≥3 are selected globally. When the numbers of selected terms are converted into aggressivity of term reduction all three collections show approximately 68% (see Table 1).

Table 1. Term selection and aggressivity of term reduction

Test Collections								
Initial Set			Recat-1			Recat-2		
No. of Terms	DF≥3	ATR	No. of Terms	DF≥3	ATR	No. of Terms	DF≥3	ATR
6,796	2,156	68%	6,787	2,171	68%	5,738	1,914	67%

ATR is an acronym for Aggressivity of Term Reduction

The selected terms are weighted using TF*IDF and documents are represented as vectors of weighted terms. A test collection is generally split into a training set that trains a classifier, and a test set that evaluates the performance of the classifier. And, sometimes a validation set (a.k.a., hold-out set or tuning set) is separately maintained for optimizing parameters. Our test collection is randomly divided into a training set

(80%) and a test set (20%). In our experiments, a validation set to choose k value is not created for the reason described in Section 3.2.

The categorization performance is measured by recall and precision. In order to present the experimental results in a single measure of effectiveness, we combine the recall and precision into a single measure, called F_1 measure. Thus, experimental results are reported based on the F_1 measure.

$$F_1 = \frac{2PR}{P+R}$$

where P denotes precision and R is recall.

In our experiments, the categorization performances of the three datasets are evaluated using a kNN algorithm as shown in Fig. 1. And the performances of the two re-classified datasets are compared with the Initial set. Finally, the differences are analyzed through t-tests to determine that they have statistical significance.

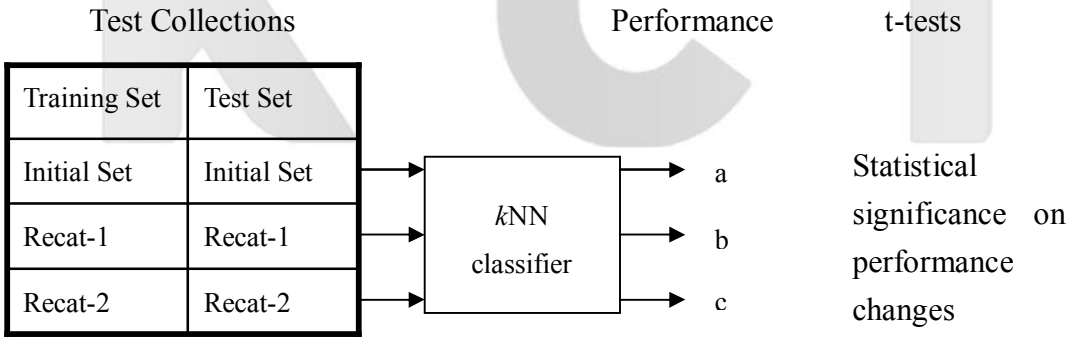

Fig. 1. Experiment model

3.2 **Text Collection**

For the experiments, a subset of the Korea Institute of Science and Technology (KISTI)'s JAFO database, which has over 14 million records as of December 2004, was chosen. The JAFO database includes journal article citations in engineerings and sciences, and the coverage ranges from May 1992 to present. We use the real world collection for our experiments instead of a standard benchmark collection because the focus of our research is to explore the real life situation. To facilitate a proper

evaluation, the corpus is created through several selection and refining processes: (1) selection of English language records with abstracts (i.e., Lang=EN, Abstract=1), (2) validity confirmations as to English language and English abstracts with computer algorithms, (3) removal of unlabeled documents and documents with invalid label(s). The resulting dataset was about 150 thousand documents. A goal of our experiments is to observe the variation of categorization performance between the initial classification group and the re-classification groups, which believed to be enhanced in the quality of classification. Therefore, we should select a reasonable size of a subset that could be manually re-classified economically. The corpus, which is a subset of the above dataset, used in the experiments consists of 1,150 documents (6 categories) in computer engineering. This corpus is divided into a training set and a test set only. Thus, we find an optimal k value from the test collection without creating a validation set.

The reason for running all the k values on the test collections is the relatively small size of the corpus. It was not large enough to split off a validation set separately.
The rationale behind trying to maintain the size of the test collections as big as possible is simple. While it has been perceived that it is necessary to have a certain number of positive training examples per category to elicit the best or an optimal categorization performance (Cai & Hoffman 2003; Joachims 1999; Brank et al. 2002; D'Alseesio et al. 1998; Ruiz & Srinivasan 1999), there has been no answer for the minimal number of training examples needed in a category. Until recently, we have only a limited knowledge that too small number of documents per category makes the interpretation of performance difficult due to high variance (Bennett 2003; Lewis et al. 1996). Only lately it was found that at least over 100 training examples per category is needed to earn an optimal performance in text categorization (Shim 2006). Thus, we tried to maintain near 100 documents per category in our experiments. This was particularly necessary because after re-classification in the resulting sets, the Recat-1 set and the Recat-2 set, the document distribution became so skewed (see Section 4.1).

Before re-classifying the dataset manually, duplications are examined. Among 1,150 documents 30 documents (15 pairs) are turned out to be duplicate documents. Only 3 pairs of duplicate documents have identical categories in each pair, and the rest of the document-pairs have different categories in the pairs (the same documents have different terms). Overall, among the duplicate documents indexing inconsistency rate reached 80% (see Table 2).

Table 2. Inconsistency rate in duplicate documents

No. of Duplicates	No. of Dups. with Category Inconsistency	Inconsistency Rate
30 docs (15 pairs)	24 docs (12 pairs)	80%

3.3 A *k*NN Classifier

A classifier is a key component of text categorization. It learns the "flavor" of texts and assigns new documents to appropriate categories. There are many classifier algorithms such as *k*NN, SVM, Rocchio, Naïve Bayes, Neural Network, and so on. Each model has different versions and sometimes several algorithms are used as a group (called classifier committees or ensembles). In our experiment a *k*NN algorithm is used for classification since *k*NN methods have shown to be very effective experimentally (Joachims 1998; Lam, and Ho 1998; Yang, and Liu 1999), while the basic idea is remarkably simple and easy to understand.

The basic idea of *k* nearest neighbor algorithms is that if there is the most similar document in the training set, then the obvious decision is to assign the new document in the same category. Typically more than one nearest neighbor is used for the decision because more documents provide better basis for the robust decision.

To find the nearest neighbors, a measure of similarity needs to be defined. Usually the Euclidean distance between documents or the cosine coefficient in the vector space is used as a similarity measure. Here, cosine coefficient is used as a similarity measure between training documents and input documents in order to extract *k* nearest neighbors:

$$S(d_i, d_j) = \frac{\sum_{k=1}^{t} w_{ki} \cdot w_{kj}}{\sqrt{\sum_{k=1}^{t} w_{ki}^2} \cdot \sqrt{\sum_{k=1}^{t} w_{kj}^2}}$$

where d_i denotes input documents, d_j training documents, w_{ki} and w_{kj} are weights of terms appeared in the vectors of d_i and d_j, respectively.

Once the *k* nearest neighbors are identified by sequential comparisons using the similarity measure, it is necessary to combine the scores in some way to make categorization decisions. Methods of making category decisions include basically a majority voting algorithm, which takes a category that appears most frequently among

the set of k nearest neighbors, a similarity score summing algorithm, which chooses a category that scores the highest similarity when summing up the similarity scores of documents belong to each category of k nearest neighbors, and hybrid algorithms that incorporate similarity scores and category frequency. In our experiment more sophisticated algorithm is used. Different from the previous methods, this algorithm takes the conditional probability of input document that might be assigned to a category into account. Thus, relevance score of each category is estimated as the weighted sum of the conditional probabilities (Yang 1994).

$$rel(c_k, d_i) = \sum_{d_j \in (k\_top\_ranking\_documents)} sim(d_i, d_j) \times P(c_k \mid d_j)$$

where $P(c_k \mid d_j) = 1$ if a neighboring document d_j is classified in category c_k, otherwise $P(c_k \mid d_j) = 0$.

Generally, in kNN algorithms the selection of optimal k value remains empirical. It is usually computed on a validation set and the k value yields the highest categorization effectiveness is selected. In previous research, Kim et al. (1999) report that k=13 is the optimal value, Lee (2003) also finds k=13 is the best when experimenting only three values of k=5, 10, 13. On the other hand, Larkey and Croft (1996) experiment with relatively bigger values of k=20 and Yang (1994, 1999) finds $30 \le k \le 45$ range yields the best performance. By summarizing various kNN experiments, Sebastiani (2002) concludes that increasing the value of k does not significantly degrade the performance. It appears that the optimal value of k depends on the characteristics of test collections, performance measures and so on, and there is no ultimate k value for all situations. However, k values do have unique patterns. According to Jackson, and Moulinier (2002), in general the value of k depends upon the following two aspects:

1. How close the classes are in the feature space. The closer the classes, the smaller k should be.
2. How typical the training documents are in a given class. If they are very heterogeneous, then a larger k is appropriate to ensure a representative sample (p. 149).

In our experiment, the best value of k for each dataset are selected comparing the performance of k=1~50 on the three datasets instead of a validation set.

4. Results and Analysis

4.1 **Result of Manual Re-Classification**

Although less than perfect categorization accuracy was expected, the observation at the end of Section 3.2 suggests that the problem of indexing inconsistency is more serious in our corpus. Therefore, we believe that we have more chance to enhance the quality of pre-assigned subject categories by re-classifying the initial set.

For re-classification, the 1,135 documents were split into 567 and 568 documents sets, and the list of each set was assigned to two groups of 4 information specialists. Four specialists group into two teams so that inter-indexer inconsistency could be minimized. And, in addition to working as teams in order to maximize the objectivity of classification decisions, the team members are instructed to consult the full papers on the web before making classification decisions.

During the re-classification process it is found that some documents are outside the scope of the 6 categories in computer engineering. For example, documents might belong to Electronic/Electric Engineering, Ceramic Engineering, and Industrial Physics are wrongfully included in the corpus. However, those areas are so closely related to computer engineering that different indexers or different systems according to their principles and service policies might have different opinions on our judgement about this matter. Particularly, "inter-system inconsistency," which denotes that many systems overlap in their collections of documents but may use different terms following their principles and service policies, might create indexing inconsistency like the one above. For example, a library serves for a computer engineering institution may have an indexing policy to put above documents among computer materials. This is essentially a reasonable inconsistency.

Thus, we decide to generate two sets of re-classified sets: Recat-1 and Recat-2. The Recat-1 is a re-classified set of the Initial set and basically reshuffles it among the given 6 categories, and the Recat-2 is smaller one with the documents that judged not to belong to the 6 categories removed from the Recat-1. The distribution of documents per categories in the initial set and re-classified sets are shown in Table 3. In the Recat-1, 15 duplicate documents from the Initial set are removed, and in the Recat-2, 202 documents judged not to belong to computer engineering are eliminated from the Recat-1.

Table 3. No. of documents/category in Initial set, Recat-1, and Recat-2

	Category (subject code)	# of Docs. in Initial Class.	Recat-1	Recat-2
1	Computers, General (ET0600)	185	1	1
2	Computer Theory (ET0602)	214	105	97
3	Computer Hardware (ET0604)	198	154	38
4	Computer Software (ET0606)	196	95	95
5	Applied Computer Technologies (ET0608)	191	708	632
6	Information Processing (ET0610)	166	72	70
	Total No. of Documents	1,150	1,135*	933**

* 15 duplicates are removed; ** 202 documents judged other than computer engineering are eliminated

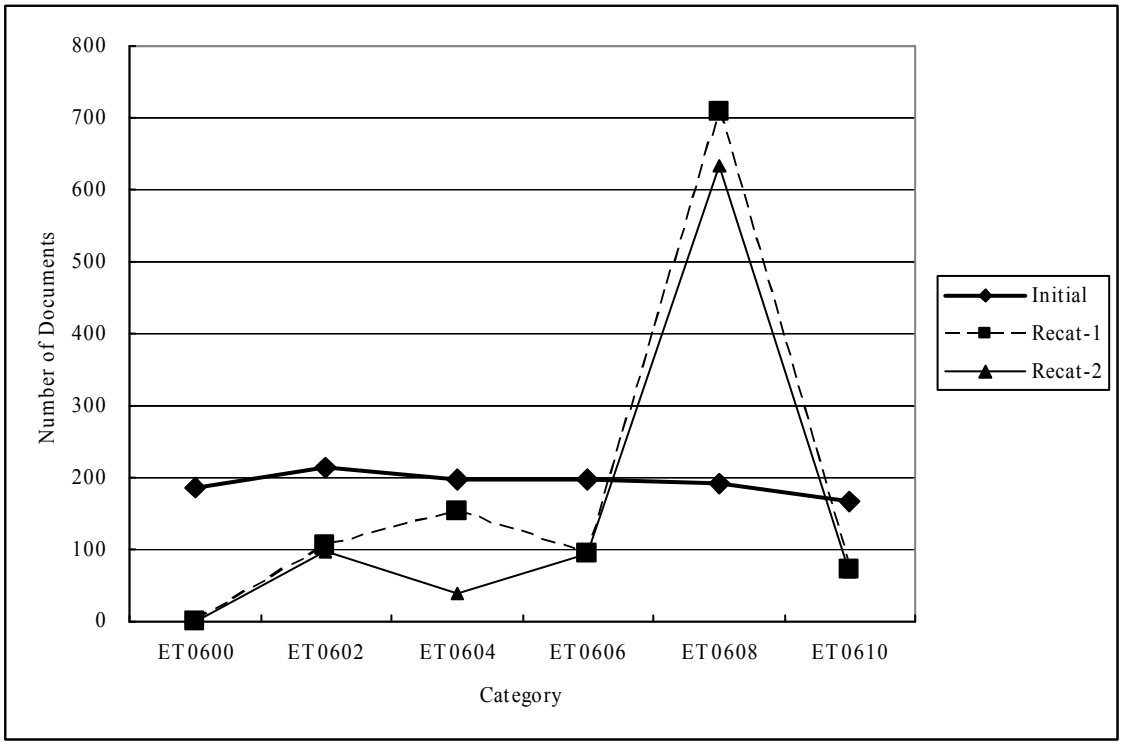

Fig. 2. Distribution of documents per category in Initial set, Recat-1, and Recat-2

As shown in Fig. 2, the Initial set shows even distribution among categories, but both the Recat-1 and the Recat-2 exhibits extreme skewness. This proves that the quality of pre-assigned categories in the Initial set was very low. To find out the classification accuracy of the Initial set, the number of documents assigned to each category in the Initial set and the number of documents remained in the same category after re-classification are compared in Table 4. On the average overall classification accuracy of the Initial set is only 16%, which is far below the indexer inconsistency rates reported

in previous research. More detailed descriptions and diagrams concerning re-assignment of documents among the categories after re-classification are described in Shim (2006).

Table 4. Classification accuracy

	Category (subject code)	# of Docs in Initial Class./ (Dups.)	# of Docs Remained	Class. Accuracy
1	Computers, General (ET0600)	185 (5)	0	0%
2	Computer Theory (ET0602)	214 (0)	14	7%
3	Computer Hardware (ET0604)	198 (5)	31	16%
4	Computer Software (ET0606)	196 (2)	21	11%
5	Applied Computer Technologies (ET0608)	191 (3)	117	61%
6	Information Processing (ET0610)	166 (0)	5	3%
	Total No. of Documents	1,150 (15)	188	16%

* For calculation of accuracy, the 15 duplicate documents are considered as incorrect assignments

The three datasets are divided into training set and test set. The following tables show the number of documents in each category and the split of training set and test set in the three datasets.

Table 5. Training/Test set split in Initial Set

	Category (subject code)	# of Docs.	Training Set (80%)	Test Set (20%)
1	Computers, General (ET0600)	185	148	37
2	Computer Theory (ET0602)	214	171	43
3	Computer Hardware (ET0604)	198	158	40
4	Computer Software (ET0606)	196	157	39
5	Applied Computer Technologies (ET0608)	191	153	38
6	Information Processing (ET0610)	166	133	33
	Total No. of Documents	1,150	920	230

In the Recat-1, almost all the documents in the 'Computers, General' category are relocated to other categories and the remaining document is not able to constitute the

training set and test set. The numbers in the parentheses indicate the number of documents increased or decreased from/to the category as a result of re-classification. And, in this dataset since the 15 duplicate documents are removed the 'Total No. of Documents' is less than 15 documents compared to that of the Initial set (see Table 6).

Table 6. Training/Test set split in Recat-1

	Category (subject code)	# of Docs.	Training Set (80%)	Test Set (20%)
1	Computers, General (ET0600)	1 (-184)	N/A	N/A
2	Computer Theory (ET0602)	105 (-109)	84	21
3	Computer Hardware (ET0604)	154 (-44)	123	31
4	Computer Software (ET0606)	95 (-101)	76	19
5	Applied Computer Technologies (ET0608)	708 (+517)	566	142
6	Information Processing (ET0610)	72 (-94)	58	14
	Total No. of Documents	1,135*	907	227

* 15 duplicate documents are eliminated
( ) numbers in parentheses indicate the numbers of documents increased/decreased after re-classification

The Recat-2 is a dataset that eliminates the documents judged not belong to the given 6 categories of computer engineering. The numbers in the parentheses indicate the number of documents removed from each category following the judgements. The resulting size is the smallest among the three datasets. As a consequence, document frequency per category in the training set is also shrunk to some degree (see Table 7). This might influence the categorization performance even we believe that this dataset has the highest accuracy in classification. The ground for this speculation is the Shim's finding (2006) that at least over 100 training examples per category is needed to earn an optimal performance in text categorization.

Table 7. Training/test set split in Recat-2

	Category (subject code)	# of Docs.	Training Set (80%)	Test Set (20%)
1	Computers, General (ET0600)	1	N/A	N/A
2	Computer Theory (ET0602)	97 (-8)	78	19
3	Computer Hardware (ET0604)	38 (-116)	30	8
4	Computer Software (ET0606)	95 (-0)	76	19
5	Applied Computer Technologies (ET0608)	632 (-76)	506	126

6	Information Processing (ET0610)	72 (-2)	56	14
	Total No. of Documents	933* (-202)	746	186

* 202 documents other than computer engineering are eliminated
( ) numbers in parentheses indicate the numbers of documents removed judged not to belong to the six categories

4.2 **Evaluation of Text Categorization Performances**

We have conducted experiments on three datasets, the Initial set, the Recat-1, and the Recat-2. All documents for training and testing go through a pre-processing step. We use DF as the term selection criterion and TF * IDF as the weighting function for text representation in our experiments.

The overall performances of the 3 datasets are evaluated using the macroaveraged F_1. The macroaveraged F_1 computes the F_1 values for each category and then takes the average over the per-category F_1 scores (for detailed description of averaging methods, refer to Sebastiani 2002, p.33).

In our experiments, k=1~50 values have been tried to find the optimal k values on each of the above three datasets. The optimal values of k are 26 for the Initial set, 19 for the Recat-1, and 30 for the Recat-2. The k values that yield highest performances on the Initial set actually include 26, 34, 39, 40, 41, 48, and the smallest value of the k=26 is chosen because of the computational efficiency. The average of three datasets' performances reaches the peak at k=19, 21, 26 and slowly deteriorates with the growth of k value (see Fig 3).

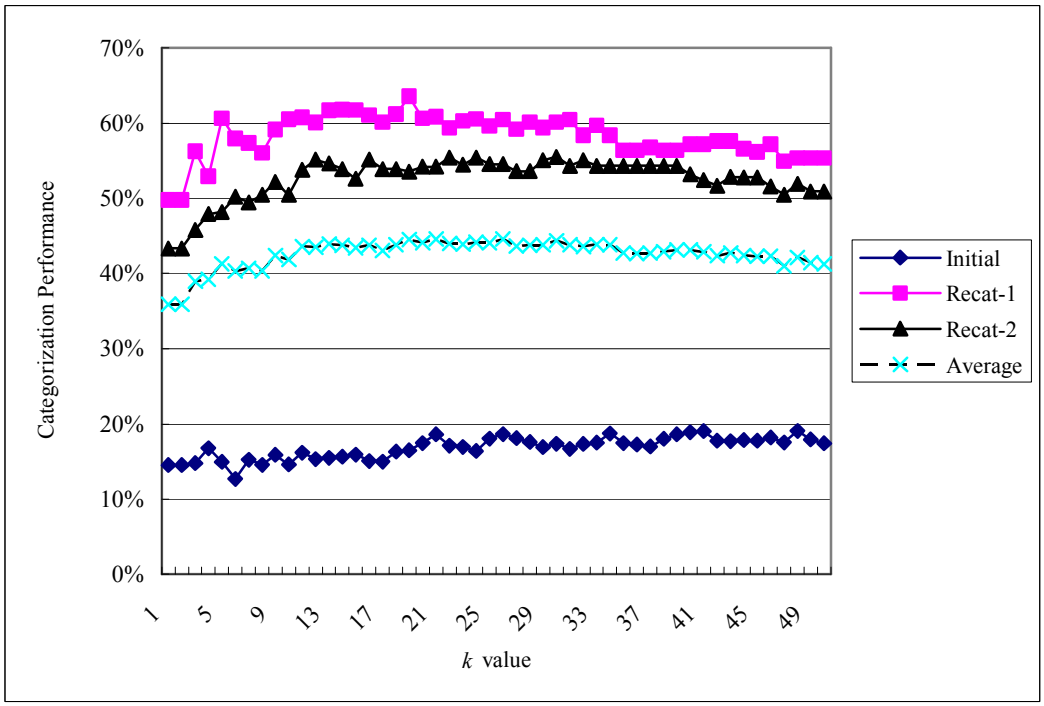

Fig. 3. Comparison of the performances of k values

In Table 8, the macroaveraged F_1 along with precision and recall of the three datasets is presented. It is clear that as expected the macroaveraged F_1 of the Recat-1 outperforms that of the Initial set. The performance of the former is 3.6 times higher than that of the latter. On the other hand, the macroaveraged F_1 of the Recat-2, which we expecte to show the highest performance, enhances the categorization performance of the Initial set only 3 times (see Table 8 and Fig. 4). The macroaveraged performances of the Recat-1 and the Recat-2 include only 5 of the given 6 categories. It is because when the Initial set is re-classified, the number of documents left in the 'Computer, General (ET0600)' category is only one, and the categorization performances of the Recat-1 and the Recat-2 are not able to be measured. Instead, in the 'Initial set' column of Table 8., the performance of the Initial set is averaged with and without the category both to facilitate "fair" comparisons. The F_1 values in the parentheses indicate the macroaveraged F_1 of the five categories

Table 8. Comparison of the macroaveraged performances on 3 datasets

	Initial set (k=26)	Recat-1 (k=19)	Recat-2 (k=30)
F_1	17% (17%)	61%	51%

Precision	23% (24%)	76%	68%
Recall	16% (17%)	55%	47%

F_1 values in the parentheses are macroaveraged performance without ET0600 category

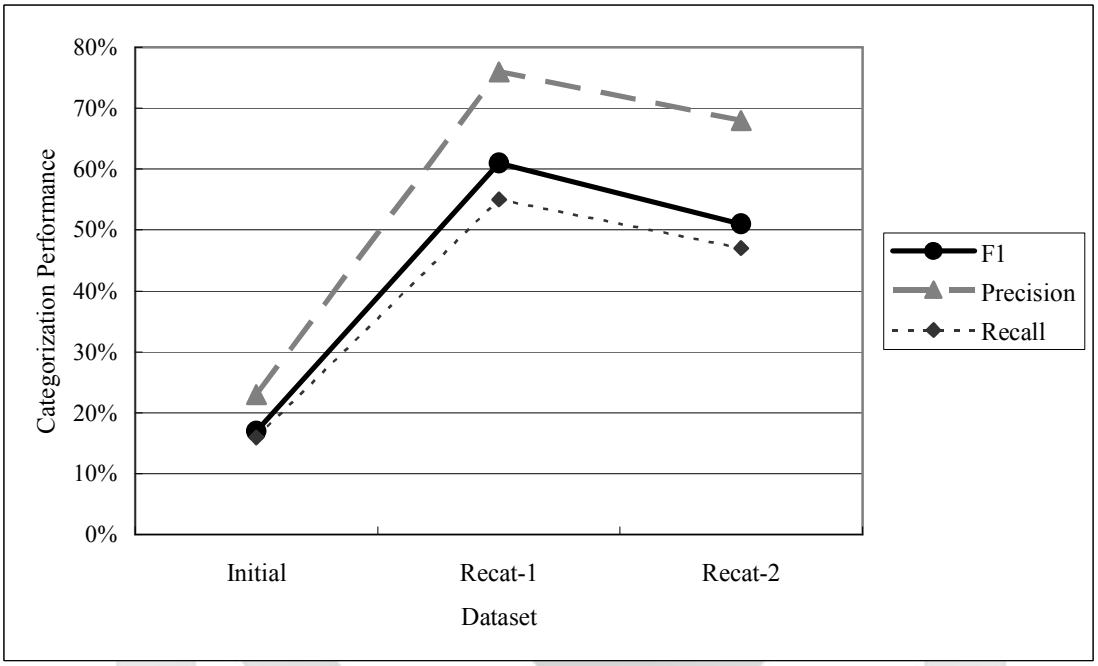

Fig. 4. Comparison of the performances of *k* values

The statistical significance of the improvement in effectiveness between the Initial set, and the Recat-1 and the Recat-2 dataset is compared using t-test. The results of the t-test confirm that the differences between the Initial set, and the Recat-1 and Recat-2 are statistically significant (see Table 9).

Table 9. Summary of the t-tests

	Initial set
Recat-1	$p < .01$ ($p < .01$)
Recat-2	$p < .01$ ($p < .01$)

The above experimental results show that the categorization performance of the Recat-2 is lower than that of the Rcat-1. We believe that the quality of classification in the Recat-2 is higher than the Recat-1 and, subsequently, the latter would yield the better performance. If the classification quality of the Recat-2 is better than the Recat-

1, we have a solid ground for this belief since we have already found that the better classification quality leads to the better performance above.

A further analysis is conducted on the category level of performance to explore whether there is a clue for explaining the "believe-to-be" discrepancy. We suspects whether the decreased performances of certain categories in the Recat-2 such as ET0602, ET0604, and ET0606 (see Fig. 5) might be due to the considerable shrinkage in the size of training examples.

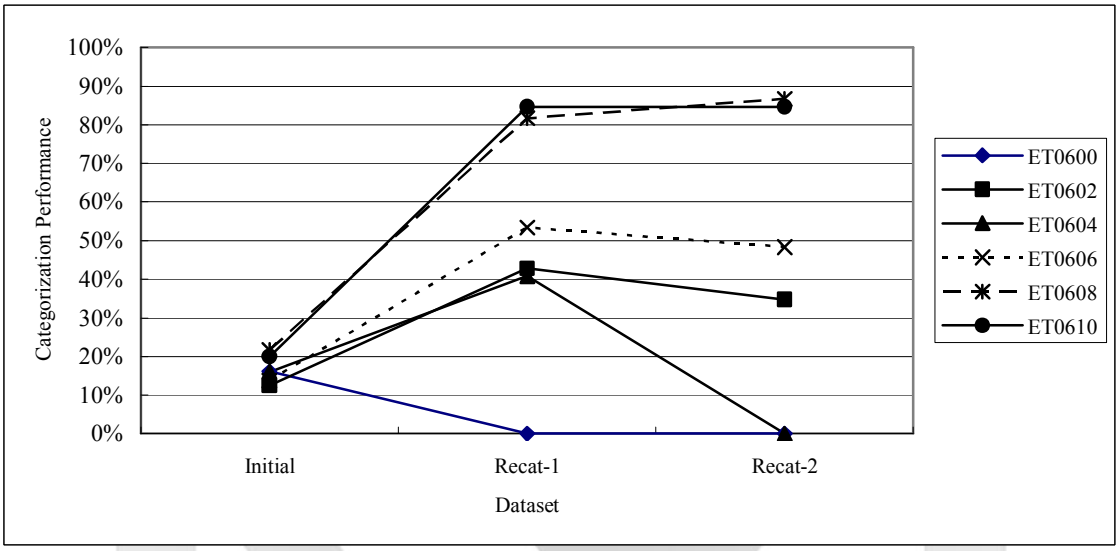

Fig. 5. Comparison of the performances of each category on 3 datasets

The performance and the number of training documents of each category in the two datasets are compared side by side in Table 10. Those categories do not indicate any systematic pattern in the variations. Only the ET0604 category with 38 documents left after losing over 100 documents during the re-classification process appears to reflect the Shim's finding (2006) mentioned in Section 3.4. The rest of the two categories do not exhibit much decline neither in the numbers of documents nor the performances. Therefore, our effort to figure out the reasons for the performance deterioration in the Recat-2 remains largely inconclusive.

Table 10. Comparison of the performances of each category on 3 datasets

Category	Initial set		Recat-1		Recat-2	
	# of Docs	F_1	# of Docs	F_1	# of Docs	F_1

ET0600	185	16%	1	--	1	--
ET0602	214	12%	105	43%	97	35%
ET0604	198	16%	154	41%	38	0%
ET0606	196	14%	95	53%	95	48%
ET0608	191	22%	708	82%	632	87%
ET0610	166	20%	72	85%	70	85%
MacroAve.	1,150	17%	1,135	61%	933	51%

5. Conclusions

In this paper, we present an investigation of imminent but seldom attacked problems in text categorization: (1) the quality of pre-assigned categories in real-world collections; (2) the relationship between the quality of classification and categorization performance. Our experiments observe the level of classification quality in a subset of a real-world collection, and manually reclassify it in order to improve the quality of classification. We evaluate the text categorization performance before and after the re-classification using a *k*NN algorithm.

The major findings of this study are as follows:

(1) In a real-life situation, the average correctness of manually assigned subject categories is 16%, and the level of categorization performance is around 17% in F_1 that is considerably lower than those from standard benchmark collections.

(2) When the quality of pre-assigned subject categories is enhanced through manual re-classification, the performance goes up to 61%.

In conclusion, by improving the correctness of pre-assigned subject labels in training documents, the categorization performance could be notably raised.

The experimental results of our investigation have some limitations to generalize. First, the size of the dataset sampled for the experiments may not be large enough in terms of the number of categories and documents. Second, the subject field of our test collection is restricted to only one domain, i.e., computer engineering. Third, only one classifier, a *k*NN algorithm, is applied. Further research is required with a larger dataset involving more categories and various subject areas, and different types

of classifiers.

References

Apté, C., F. Damerau, and S. M. Weiss. 1994. "Automated learning of decision rules for text categorization." *ACM Transactions on Information Systems*, 12(3): 233-251.

Bates, M. J. 1986. "Subject access in online catalogs: a design model." *Journal of the American Society for Information Science*, 37(6): 357-376.

Bennett, P. N. 2003. "Using asymmetric distributions to improve text classifier probability estimates." In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 111-118.

Blair, D. C. 1986. "Indeterminancy in the subject access to documents." *Information Processing & Management*, 22(2): 229-241.

Brank, J., M. Grobelnik, N. Milić-Frayling, and D. Mladenić, 2002. "Feature selection using linear support vector machines." In *Proceedings of the 3rd International Conference on Data Mining Methods and Databases for Engineering*.

Cai, L. and T. Hofmann. 2003. "Text categorization by boosting automatically extracted concepts." In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 182-189.

Chen, H., A. K. Danowitz, K. J. Lynch, E. E. Goodman, and W. K. McHenry. 1994. "Explaining and alleviating information management indeterminism: a knowledge-based framework". *Information Processing & Management*, 30(4): 557-577.

Cleverdon, C. 1984. "Optimizing convenient online access to bibliographic databases." *Information Services and Use*, 4(1): 37-47.

Chung, Y-M. 2005. 『정보검색연구 [*Research in information retrieval*]』. Seoul: KuMee Trade.

D'Alessio, S., K. Murray, , and R. Schiaffino. 1998. "The effect of using hierarchical classifiers in text categorization." In *Proceedings of the 3rd Conference on Empirical Methods in Natural Language Processing*, 1-18.

Guthrie, L., J. Guthrie, and J. Leistensnider. 1999. "Document classification and routing". In *Natural language information retrieval*, edited by T. Strzalkowski, 289-310. Boston: Kluwer Academic Publishers.

Hooper, R. S. 1965. *Indexer consistency test: origin, measurement, results and utilization*. Bethesda, MD: IBM Corporation.

Hurwitz, F. I. 1969. "A study of indexer consistency". *American Documentation*, 20: 92-94.

Jackson, P. and I. Moulinier. 2002. *Natural language processing for online applications*: *text retrieval, extraction and categorization*. Amsterdam: John Benjamins Publishing Company.

Jacoby, J. and V. Slamecka. 1962. *Indexer consistency under minimal conditions*.

Bethesda, MD: Documentation, Inc.

Joachims, T. 1999. "Transductive inference for text classification using support vector machines." In *Proceedings of ICML-99: 16th International Conference on Machine Learning*, 200-209.

Joachims, T. 1998. "Text categorization with support vector machines: learning with many relevant features." In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, 137-142.

Kim, S. B., B. H. Yoon, D. H. Baek, K. S. Han, and H. C. Lim. 1999. 『문서범주화를 위한 선형분류기와 kNN의 결합모델 [*Combining a linear classifier and a kNN model for text categorization*]』. In *1999 Spring Symposium of Korean Society for Cognitive Science*, 225-231.

Lam, W. and C. Y. Ho. 1998. "Using a generalized instance set for automatic text categorization." In *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, 81-89.

Larkey, L. S. and W. B. Croft. 1996. "Combining classifiers in text classification." In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 289-297.

Lee, H. W. 2003. 『복합분류기를 이용한 웹 문서범주화에 관한 실험적 연구 [*An experimental study on categorization of web documents using an ensemble classifier*]』. MA thesis, Yonsei University.

Leininger, K. 2000. "Interindexer consistency in PsycINFO." *Journal of Librarianship and Information Science*, 32(1): 4-8.

Lewis, D. D. and W. A. Gale. 1994. "A sequential algorithm for training text classifiers." In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 3-12.

Lewis, D. D., R. E. Schapire, J. P. Callan, and R. Papka. 1996. "Training algorithms for linear text classifiers." In *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, 298-306.

Ruiz, M. E. and P. Srinivasan. 1999. "Combining machine learning and hierarchical indexing structures for text categorization." In *Proceedings of the 10th ASIS SIG/CR Classification Research Workshop*, 107-124.

Saracevic, T. 1991. "Individual differences in organizing, searching and retrieving information". In *Proceedings of the 54th Annual Meeting of the Society for Information Science*, 82-86.

Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1): 1-47.

Shim, K. 2006. 『학습문헌집합의 속성에 따른 문헌 범주화 성능 실험 [*An experimental study ascertaining the relationships between the characteristics of a training document set and the performance of text categorization*]』. Ph.D. diss., Yonsei University.

Van Rijsbergen, C. J. 1979. *Information retrieval*. 2nd ed. London : Butter Worths.

Weiss, S. M., C. Apté, F. J. Damerau, D. E. Johnson, F. J. Oles, T. Goetz, and T. Hampp. 1999. "Maximizing text-mining performance." *IEEE Intelligent Systems*, 14(4): 63-69.

Weiss, S. A., S. Kasif, and E. Brill. 1996. "Text classification in USENET Newsgroups: a progress report". In *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*.

Yang, Y. 1999. "An evaluation of statistical approaches to text categorization." *Information Retrieval*, 1: 69-90.

Yang, Y. 1994. "Expert network: effective and efficient learning from human decisions in text categorization and retrieval." In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, 13-22.

Yang, Y. and X. Liu. 1999. "An re-examination of text categorization methods." In *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, 42-49.