Sampling Strategies and Learning Efficiency in Text Categorization

Yiming Yang
Section of Medical Information Resources, Mayo Clinic/Foundation
Rochester, Minnesota 55905 USA
yang@mayo.edu

Abstract

This paper studies training set sampling strategies in the context of statistical learning for text categorization. It is argued sampling strategies favoring common categories is superior to uniform coverage or mistake-driven approaches, if performance is measured by globally assessed precision and recall. The hypothesis is empirically validated by examining the performance of a nearest neighbor classifier on training samples drawn from a pool of 235,401 training texts with 29,741 distinct categories. The learning curves of the classifier are analyzed with respect to the choice of training resources, the sampling methods, the size, vocabulary and category coverage of a sample, and the category distribution over the texts in the sample. A nearly-optimal categorization performance of the classifier is achieved using a relatively small training sample, showing that statistical learning can be successfully applied to very large text categorization problems with affordable computation.

Introduction

Assigning predefined categories to free texts, *text categorization*, has wide application in practical databases. For example, insurance codes must be assigned to diagnoses in patient records at the Mayo Clinic for billing and research purposes. Documents in MEDLINE, one of the world's largest, on-line bibliographical databases, are indexed using subject categories for the purposes of retrieval. It costs the National Library of Medicine about two million dollars each year to manually index MEDLINE documents (about a third of a million new entries a year), and the Mayo Clinic about 1.4 million dollars annually for manual coding of diagnoses (over two million diagnoses a year). Automatic or semi-automatic text categorization tools can help reduce these costs and may also improve accuracy and consistency.

Computer-based text categorization technologies include:

- *naive word-matching* (Chute, Yang, & Buntrock 1994) which matches categories to text based on the shared words between the text and the names of categories;
- *thesaurus-based matching* (Lindberg & Humphreys 1990) which uses lexical links (constructed manually or automatically in advance) to relate a given text to the names or descriptive phrases of categories; and
- *empirical learning of term-category associations* from a training set of texts and their categories assigned by humans.

Naive word-matching is the weakest method. It cannot capture any categories which are conceptually related to a text but happen not to share "the right words" or any words with the text. Thesaurus-based matching suffers from the weakness that the lexical links are typically static, not sensitive to the context in which a term (word or phrase) is used, and hence fails when the meaning of a term is context-dependent. If the thesaurus is hand-built, it also suffers from high cost and low adaptivity across domains. Empirical learning from categorized texts, on the other hand, fundamentally differs from word-matching because it is based on human relevance judgments, statistically capturing the semantic associations between terms and categories. Moreover, most empirical learning formalisms offer a context sensitive mapping from terms to categories, for example, decision tree methods (Quinlan 1986) (Lewis 1991), Bayesian belief networks (non-naive ones) (Tzeras & Hartman 1993), neural networks (Schütze, Hull, & Pedersen 1995), nearest neighbor classification methods (Creecy *et al.* 1992), (Masand, Linoff, & Waltz 1992) (Yang 1994), and least-squares regression techniques (Fuhr, Hartmanna, & et al. 1991) (Yang & Chute 1994).

While empirical learning holds great potential for high accuracy text categorization, few practical systems have developed due to difficulties in scaling to large problems. The MEDLINE database, for example, uses about 17,000 subject categories (Nat 1993)

to index its articles. The Mayo Clinic, as another example, uses 29,741 categories (Com 1968) to code diagnoses. The largest number of categories ever tried in a decision tree or a neural network, however, is only a few hundreds or less (Schütze, Hull, & Pedersen 1995). Bayesian belief networks have a similar scaling-up difficulty, and a "naive" version (Lewis 1991) is often used for computational tractability. Naive Bayesian methods assume term independence in category prediction, fundamentally sacrificing the strength of the original framework in handling the context sensitivity of term-to-category mapping. In contrast, nearest neighbor approaches and linear regression methods require less computation, and have been applied to relatively large categorization problems (Creecy *et al.* 1992), (Masand, Linoff, & Waltz 1992) (Yang 1994) (Fuhr, Hartmanna, & et al. 1991) (Yang & Chute 1994). Nevertheless, for large problems with tens of thousands of categories, the learning effectiveness and the computational tractability of empirical methods remains a largely unexplored area.

Sampling strategies are important for both the effectiveness and the efficiency of statistical text categorization. That is, we want a training set which contains sufficient information for example-based learning of categorization, but is not too large for efficient computation. The latter is particularly important for solving large categorization problems in practical databases[1]. The amount of available training data is often nearly infinite. For example, there are roughly seven million documents accumulated in MEDLINE, and a similarly large number of diagnoses at Mayo. All of these documents and diagnoses have manually assigned categories, and are thus eligible as training data. However, using all the available data for training is computationally impractical, and may also be unnecessary. Statistical sampling theories show that one can obtain a fairly accurate estimate of a population parameter using a relatively small sample. In this paper, I focus on the following questions regarding effective and efficient learning of text categorization:

- Which training instances are most useful? Or, what sampling strategies would globally optimize text categorization performance?
- How many examples are needed to learn a particular category? How can one balance the local optimality (for individual categories) versus global optimality?
- How can one measure learning efficiency to optimize the trade-off between categorization accuracy and learning efficiency?
- Given a real-world problem, how large a training sample is large enough?

[1] In contrast, traditional inductive machine learning systems are trained with fewer examples, typically dozens or hundreds, where most examples in such small training sets prove necessary, and hence sampling is less important.

Methodology

Training Data and a Learning System

To study sampling strategies, one needs a pool of training data from which samples can be drawn, and a classification system against which the effects of different strategies can be tested and compared. In this paper, the categorization of Mayo diagnoses (DXs) will be the problem domain, and a nearest neighbor classifier, named Expert Network or ExpNet, the system. For convenience, I use *text* to refer to either a DX or a category definition phrase. Texts in our system are represented using the conventional vector space model (Salton 1989). That is, a text is represented as a vector whose dimensions are unique words in a diagnosis collection, and whose elements are word weights in this text. A word is typically weighted using the product of the inverse document frequency in a training collection, and the word frequency in the text.

ExpNet uses a collection of DXs with human assigned categories as a training set. It generates a ranked list of candidate categories for a new diagnosis based on the categories of its nearest neighbors in the training diagnoses. Given a new, unlabelled DX, the system searches for its k nearest neighbors (k-NN) among the training DXs. The cosine of the angle between a training DX (represented as a vector) and the new DX is used to measure the similarity of the training DX. The training DXs with the k highest cosine values are chosen as the nearest neighbors, and their categories are used to predict the categories of the new DX. The relevance score of a category with respect to the new DX is estimated by the weighted sum of the categories of the k-NN, where the weights are the cosine-similarity scores of these k-NN. ExpNet does not require any off-line training, but does require an on-line search for the k-NN of each new text, which has a time complexity linear in the size of the training set (the number of training texts). See (Yang 1994) for additional details and for a discussion on the choice of the parameter k.

A nearest neighbor classifier was selected for this study because of its relatively low computation cost and the relative ease with which it scales to large problems. Also, nearest neighbor classification performs at least as well as least squares fit (Yang 1994) or rule-based approaches (Creecy *et al.* 1992). A production

version of ExpNet was developed at the Mayo Clinic, and is in daily use as a search engine in computer-aided human coding of patient records (Chute, Yang, & Buntrock 1994).

The pool of training data consists of a subset of diagnoses from patient records at Mayo, and the definition phrases of the categories in HICDA-2 (Hospital Adaptation of ICDA, 2nd Edition)(Com 1968). About 2.4 million diagnoses (DXs) are coded each year using the HICDA-2 categories. However, it would be impractical to use all of the accumulated DXs for training. ExpNet operates as an on-line category searcher. When a coder types in a new DX, ExpNet provides a ranked list of potential categories for the user to select. The on-line response time is proportional to the number of unique DXs in the training set (Yang 1994). With a quarter million training DXs, the response time is about 1.5 seconds per new DX when running ExpNet on a SPARCstation 10. Five million training DXs would increase the response time to 32 seconds per DX; 10 million training DXs would increase the response time to more than one minute which is too slow for effective user-machine interaction. Similarly, the memory usage scales linearly with training set size.

The *pool* chosen for this study is the same training set used by our current production system, which consists of 205,660 DXs from patient records (a few months accumulation in the period from October 1993 to March 1994) and 29,741 category definition phrases in HICDA-2. There are a total of 235,401 training texts in the pool, with 15,994 distinct words and 29,741 distinct categories. I will refer to the 205,660 DXs as the DX superset, or simply the *superset*. Subsets are derived from the superset, according to different sampling strategies. A diagnosis is a fragment of free text, with 1-26 words, or 3 words per DX on average. A category definition consists of 3 or 4 words. Each category definition has a unique identifier, a *code*. A DX has 1-7 codes, or an average of 1.14 codes per DX.

Global Strategies

One rule of thumb in instance-based learning is that at least ten positive examples are required to reliably learn a category. I will use *instance* to refer to a DX/category pair (some DX's generate multiple pairs). Among the 29,741 HICDA-2 categories, 66% of them have no instance in the DX superset, 10% have one instance only, and 15% has 2-9 instances. Together, 27,078 distinct categories, or 91% of the total, have less than 10 instances in the superset. Hence, according to the rule of thumb, we do not have enough training data to learn the rare 91% of the categories. Does this mean that ExpNet is doomed to have a poor performance? Not necessarily. In fact, only a small fraction of the distinct categories is important for the global performance.

The superset contains a total of 234,465 instances of all categories. Only 20,719 instances belong to the rare categories which have less than ten instances. If ExpNet did not learn anything about the 27,078 rare categories (91% of total), the expected failures on the instances of these categories is only $20,719/234,465$, or 9% of the cases. In contrast, the ten most common categories together have 23,850 instances, i.e. more instances than the 27,078 rare categories. The 1611 most common categories (5% of total) account for 85% of the 234,465 instances. This means that if ExpNet only learned for the 5% most common categories, then the categorization success could be as high as 85%. Clearly, in terms of categorization success rate, common categories have more weight in the overall performance than rare categories. Hence it is crucial that common categories are well represented in a training sample, while missing a large number of rare categories in the training sample may have no statistically significant impact. We want to globally optimize the categorization performance. To achieve this, global control of sampling over categories is necessary.

Figure 1 illustrates the difference between two extreme sampling strategies on the DX superset: common category first versus rare category first. The horizontal axis measures the coverage of distinct categories in a training sample. A sample begins with an empty set, and is enlarged step-by-step by adding either the next most common or next most rare category depending on the sampling strategy. The vertical axis is the expected success rate of an ideal classifier given a training sample, assuming that categories covered in the training sample are prefectly learned. The success rate is just the probability that a particular instance belongs to a category covered by the training sample. This probability is estimated by the ratio of the number of instances of the categories in the training sample, divided by the number of instances of all categories in the superset.

The curves in Figure 1 are obtained by applying the two sampling strategies and interpolating the estimated success rates of selected training samples. The upper curve corresponds to the best sampling, i.e., common category first. The lower curve corresponds to the worst sampling, i.e., rare category first. Any other sampling strategies will have a success rate falling in between the two curves. A uniform sampling strategy over all categories, for example, is intermediate in performance since categories do not have a uniform

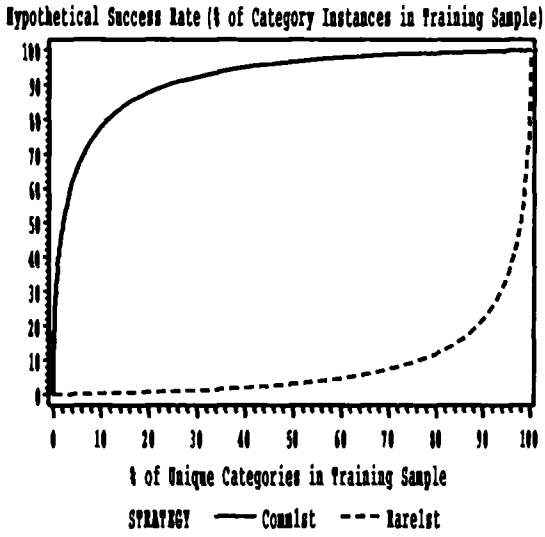

Figure 1: Learning curves of an ideal classifier on Mayo diagnoses.

distribution in the domain. Random sampling over instances, as another example, naturally favors common categories over rare categories, and hence is better than uniform sampling in a global sense. *Uncertainty sampling* (Lewis 1994) strategies are driven by the "failure" of a classifier. That is, instances that are not well classified by the learning system are added to the training sample. This results in the selection of instance belonging to rare categories that contribute little to global categorization performance. Given unevenly distributed categories, such an approach is not optimal in a global sense.

Local Strategies

While a globally optimal strategy exists in theory, it can only be approached approximately in practice since we cannot guarantee that the training sample will contain all the instances required for perfect learning. I use *local strategy* to refer to the process of selecting instances for a particular category. The questions are: how many instances are sufficient, and which instances are preferable to learn a given category? The rule of thumb suggests that one needs ten or more instances per category, but that is not a sufficient answer. Intuitively, a common category requires a larger number of instances than a rare category, because the more often a concept or category is referred to by humans, the larger number of distinct words or phrases are likely to be used. This means that a classifier would need a large number of instances to statistically learn the mapping from natural language words to the particular concept. On the other hand, not every instance contributes useful information. Duplicates, for example, may not contribute as much as unique instances.

It would be reasonable to consider the desirable category distribution over instances in a training sample to be a function of the category distribution in the underlying population. A simple choice is a linear function; i.e., keep the number of instances of a category in a training sample proportional to the number of instances of this category in the underlying population. Such a sample can be obtained using random sampling if the population is infinite, or by a systematic approach when the population is finite. More complex functions are possible, such as square-root, logarithmic, or a combination of different functions each applies to a particular group of categories. Uncertainty sampling offers an alternative approach. Instead of setting a lower bound on the number of instances needed, it selects some instances over others based on whether the classifier needs to learn about them further. In the next section, a systematic sampling is used to implement a simple linear strategy. Other local strategies will not be further discussed due to space limitations.

Two Methods for Comparison

Two alternative sampling methods were included in this study:

1) *Proportion-enforced sampling*: a systematic sampling is used to enforce that the category distribution in a training sample reflects the distribution in the pool of training data. This is essentially very similar to random sampling, except that it avoids random departures of the samples from the underlying distribution, which is quite likely for small random samples. This method globally favors common categories and also has reasonable coverage of rare categories. This strategy is driven by a splitting parameter k. When $k = 2$, for example, only half of the rare categories with one instance in the pool will be included in each of the two subsets. On the other hand, every category with two or more instances in the pool will have at least one instance in each subset; a category with 100 instances will have 50 instances in each subset. With $k = 10$, every category with ten or more instances in the pool will have at least one instance in each of ten subsets; a category with 100 instances will have 10 instances in each subset. By varying the value of k, we can observe the effects of local strategies while globally favoring common categories.

2) *Completeness-driven sampling*: prefer some instances over others if they contribute more new words

or categories to a training set. An extreme example of this would be to take the HICDA-2 definition phrases only as a training set or as the dominating part of a training set, because they contain all the unique categories, and more unique words than the DXs in the pool. Such a method ignores the natural distribution of categories in the domain, and may consequently decrease the over-all categorization effectiveness of ExpNet. A potential advantage of such a strategy is that it has better coverage of rare cases.

Empirical Validation

Performance Measures and Test Data

Classifier effectiveness is measured by the conventional ten-point average precision (AVGP) (Salton 1989) of ranked categories per test instance. This is a global performance measure similar to correct classification rate. Given a test DX, a ranked list of candidate categories is generated by ExpNet, and the precision values at recall thresholds of 10%, 20%, ... 100% are computed and averaged to obtain a single-number measure. The AVGP values of all DXs in a test set are further averaged to achieve a global measure, referred to as the *categorization accuracy*.

Training samples are judged in terms of usefulness (i.e. categorization performance) and in terms of completeness (i.e. vocabulary and category coverage). Computational efficiency is also measured. These measures are analyzed with respect to sample size and sampling strategy. Similarly, an analysis of the correlation between accuracy improvement and computation cost helps evaluate the trade-off between categorization accuracy and learning efficiency.

A test set was selected to evaluate sampling strategies. Five test sets were collected in our previous evaluation of different categorization methods (Chute, Yang, & Buntrock 1994); none of them were from the training pool. Each set consists of about 1000 DXs arbitrarily chosen from the patient records at the time of that evaluation. By checking the common words and categories of each of these test sets and the training superset mentioned before, we found that these testing sets are similar in the sense that they all have about 97-98% of the words and 96-97% of the categories covered by the training DXs. Hence, one of the five testing sets was arbitrarily chosen for this study, containing 1071 DXs, 1249 unique words and 726 unique categories.

Preprocessing was applied to training and test data for the removal of punctuation and numbers, and for changing uppercase letters to lowercase; no stemming or removal of "noise words" was applied.

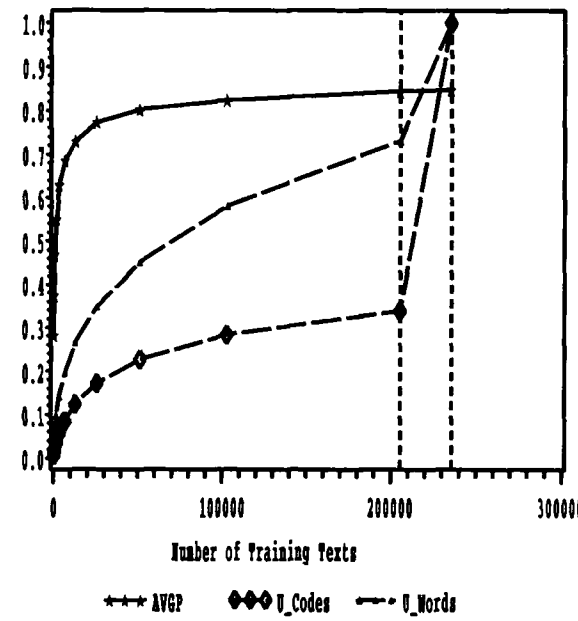

Figure 2: Learning curve of ExpNet using sort-and-split sampling.

The Results

Figure 2 shows the learning curve of ExpNet given the proportion-enforced sampling method. Training samples were derived from the superset of 205,660 DXs by selecting the first of every k instances. By setting the split parameter k to 1024, 512, ..., 4, 2, 1, samples including 200, 401, 803, ..., 205,660 DXs were obtained. These samples were used as training sets for ExpNet, and evaluated using the test set. The entire pool, i.e., the 205,660 DXs plus the 29,741 category definition phrases, were also tested as an additional larger training set. The AVGP values were computed for these training sets respectively, and plotted corresponding to the *size* (the number of texts) of the training sets. Interpolating these plots, the *learning curve* (the star-line) of ExpNet is obtained. The dashed vertical lines correspond to the the superset and the superset plus category definition phrases, respectively. The triangle-line shows the vocabulary coverage, i.e., the ratio of the number of unique words in a training set divided by the total number of unique words in the pool. The diamond-curve shows the category coverage, i.e. ratio of the number of unique categories in a training set divided by the total number of unique categories in the pool.

The interesting results are:

1) The learning curve rises rapidly when the training sets are small, and becomes relatively flat when the training set size achieves 100,000 texts or larger. This

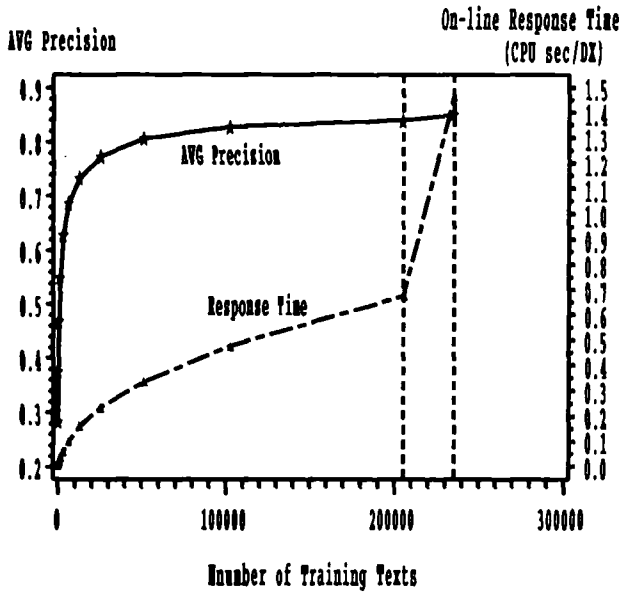

Figure 3: Trade-off between effectiveness and efficiency of ExpNet.

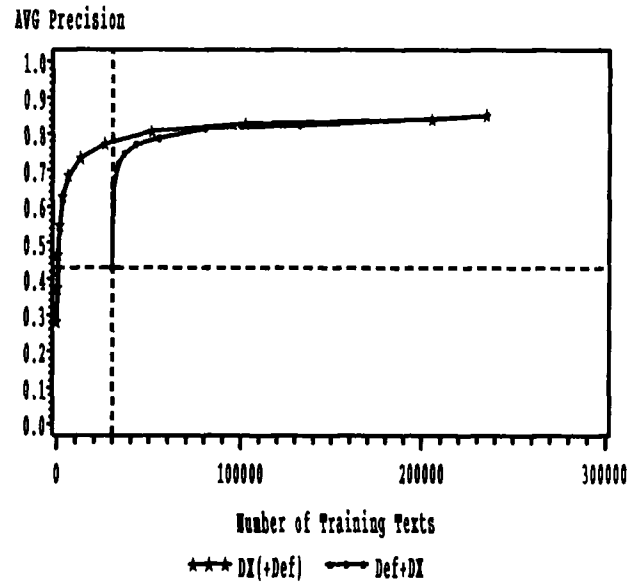

Figure 4: Learning curves of ExpNet using different sampling strategies.

suggests that high-frequency DXs are included even when the training sets are small, and that these DXs are highly influential in over-all performance. When the training sets get large, more and more rare DXs or duplicates of common DXs are included, which contribute less to global performance. Beyond the 200,000 level, further increase in the size of a training seems unlikely to have any significant improvement.

2) The slope of the unique-word curve and the unique-category curve is much larger than the learning curve, except at the very left end of this graph. This means that the improvement in word coverage and category coverage of a training set does not linearly transfer into an improvement in categorization effectiveness. In other words, a large number of words and categories are not crucial to the global categorization performance because they are rare.

3) Adding the HICDA-2 phrases to the training DXs did not improve the AVGP by much, although it significantly increased the vocabulary and category coverage of the training set. This means that most of the words and categories which are needed for classification are already included in the training DXs, and that the category definition phrases contribute little useful information to the training.

Figure 3 shows the trade-off between AVGP and the on-line response time of ExpNet. The average CPU seconds for category ranking per DX was measured. A significant time increase was observed when the HICDA-2 definition phrases are added to the training set, because most of these definition phrases are unique to the entire collection, and the response time of ExpNet is proportional to the number of unique training texts. Clearly, using HICDA-2 in addition to Mayo DXs for training doubled the computation time for only an insignificant improvement in categorization effectiveness.

Figure 4 compares the learning curve in Figure 2 with the learning curve (dot-curve) when the training sets were constructed in a different way. That is, the HICDA-2 phrases were used as the basic training set, and each of the DX subsets was added to the basic set respectively. When using the HICDA-2 phrases alone for training, the AVGP was only 42% which is similar to the performance obtained using 400-800 DXs for training, and also similar to applying word-based matching between DXs and HICDA-2 phrases (which had an AVGP value of 44% in our experiment). On the other hand, when using a similar number (25,707) of DXs instead of the 29,741 HICDA-2 phrases for training, the AVGP was 77%, or an 83% relative improvement. Clearly, given a training sample size, using the instances from the application itself is a much better choice than using category names or definition phrases. It is also clear in Figure 2 that using the category definitions in combination with DXs for training had no significant improvement over using DXs alone, if any. However, the on-line computation cost (both time and

space) is much higher when including the category definitions, because these definitions have more unique words, phrases and codes than DXs do.

In all the above experiments, we did not apply removal of "noise words" because this is not the focus of this paper. When applying word removal using a standard "stoplist" which consists of articles, prepositions, conjunctions and some common words, the AVGP on the 205,660 DX training set was improved from 84% to 86%; the on-line response of ExpNet was reduced from 1.5 to 0.7 seconds per DX.

Conclusions

This paper studied the sampling strategies for statistical learning of text categorization, using the ExpNet system on a large collection of diagnoses in Mayo patient records. The major findings include:

1) In practical databases, categories often have a non-uniform distribution over texts, which makes the "usefulness", or the global performance of a statistical classifier heavily dependent on the distribution of categories in a training sample.

2) A theoretical analysis backed by experimental evidence indicates that a global sampling strategy which favors common categories over rare categories is crucial for the success of a statistical learning system. Without such a global control, pursuing more complete coverage of words, categories or instances of a particular category could be damaging to global optimal performance, and can substantially decrease learning efficiency.

3) Using ExpNet, a globally high text categorization accuracy can be achieved by using a relatively small and "incomplete" training sample. The 86% average precision in a space of 29,741 distinct categories, with an on-line response of 0.7 second per diagnosis on a SPARCstation 10 is highly satisfactory for the current needs in our computer-assisted categorization applications.

4) The ExpNet learning curve indicates that the system achieved close to its highest accuracy on average using about 200,000 training DXs, and that significant improvement beyond this point would be difficult.

No claim is made that the particular size of an optimal or nearly-optimal training set for one application is generalizable for all applications. The optimal training set size for Mayo diagnosis categorization may not be the optimal size for MEDLINE document categorization, for example. Given that a diagnosis phrase has three words on average, and that a MEDLINE article has typically 150-200 words in its title and abstract, the necessary amounts of training data may be larger for the latter than for the former. Nevertheless, the analysis method presented here is generalizable to other domains/applications and alternative statistical classification methods.

Future research topics include:

- investigation of local sampling strategies which have not been explored in this paper, such as more complex functions of an underlying distribution in setting a lower bound on the number of needed instances, and using uncertainty sampling under the control of a globally optimal strategy;
- a sampling strategy analysis for ExpNet on different domains, such as MEDLINE documents, the Reuters newswire collection, etc.; and
- similar analyses for other statistical learning methods, e.g., the Linear Least Squares Fit mapping (Yang & Chute 1994).

Acknowledgement

This work is supported at Mayo Foundation in part by NIH Research Grants LM05714 and LM05416.

References

Chute, C.; Yang, Y.; and Buntrock, J. 1994. An evaluation of computer-assisted clinical classification algorithms. In *18th Ann Symp Comp Applic Med Care (SCAMC) JAMIA 1994;18(Symp.Suppl)*, 162–6.

Commission on Professional and Hospital Activities, Ann Arbor, MI. 1968. *HICDA-2, Hospital Adaptation of ICDA, 2nd Edition.*

Creecy, R.; Masand, B.; Smith, S.; and Waltz, D. 1992. Trading mips and memory for knowledge engineering: classifying census returns on the connection machine. *Comm. ACM* 35:48–63.

Fuhr, N.; Hartmanna, S.; and et al., G. L. 1991. Air/x - a rule-based multistage indexing systems for large subject fields. In 606-623., ed., *Proceedings of RIAO'91.*

Lewis, D. 1991. Evaluating text categorization. In *Proceedings of the Speech and Natural Language Workshop, Asilomar*, 312–31. Morgan Kaufman.

Lewis, D. 1994. A sequential algorithm for training test classifiers. In *17th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 94)*, 3–12.

Lindberg, D., and Humphreys, B. 1990. The umls knowledge sources: Tools for building better user interfaces. In *Proceedings of the 14th Annual Symposium on Computer Applications in Medical Care (SCAMC 90)*, 121–125.

Masand, B.; Linoff, G.; and Waltz, D. 1992. Classifying news stories using memory based reasoning. In *15th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 92)*, 59–64.

National Library of Medicine, Bethesda, MD. 1993. *Medical Subject Headings (MeSH).*

Quinlan, J. 1986. Induction of decision trees. *Machine Learning* 1(1):81–106.

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

Schütze, H.; Hull, D.; and Pedersen, J. 1995. A comparison of classifiers and document representations for the routing problem. In *18th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 95*, 229–237.

Tzeras, K., and Hartman, S. 1993. Automatic indexing based on bayesian inference networks. In *Proc 16th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 93)*, 22–34.

Yang, Y., and Chute, C. 1994. An example-based mapping method for text categorization and retrieval. *ACM Transaction on Information Systems (TOIS)* 253–277.

Yang, Y. 1994. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *17th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 94)*, 13–22.