A Comparison of Manual and Automatic Constructions of Category Hierarchy for Classifying Large Corpora

Fumiyo Fukumoto
Interdisciplinary Graduate
School of Medicine and Engineering
Univ. of Yamanashi
fukumoto@skye.esb.yamanashi.ac.jp

Yoshimi Suzuki
Interdisciplinary Graduate
School of Medicine and Engineering
Univ. of Yamanashi
ysuzuki@ccn.yamanashi.ac.jp

Abstract

We address the problem dealing with a large collection of data, and investigate the use of automatically constructing category hierarchy from a given set of categories to improve classification of large corpora. We use two well-known techniques, partitioning clustering, k-means and a $loss\ function$ to create category hierarchy. k-means is to cluster the given categories in a hierarchy. To select the proper number of k, we use a $loss\ function$ which measures the degree of our disappointment in any differences between the true distribution over inputs and the learner's prediction. Once the optimal number of k is selected, for each cluster, the procedure is repeated. Our evaluation using the 1996 Reuters corpus which consists of 806,791 documents shows that automatically constructing hierarchy improves classification accuracy.

1 Introduction

Text classification has an important role to play, especially with the recent explosion of readily available online documents. Much of the previous work on text classification use statistical and machine learning techniques. However, the increasing number of documents and categories often hamper the development of practical classification systems, mainly by statistical, computational, and representational problems(Dietterich, 2000). One strategy for solving these problems is to use category hierarchies. The idea behind this is that when humans organize extensive data sets into fine-grained categories, category hierarchies are often employed to make the large collection of categories more manageable.

McCallum et. al. presented a method called 'shrinkage' to improve parameter estimates by taking advantage of the hierarchy(McCallum, 1999). They tested their method using three different real-world datasets: 20,000 articles from the UseNet, 6,440 web pages from the Industry Sector, and 14,831 pages from the Yahoo, and showed improved performance. Dumais et. al. also described a method for hierarchical classification of Web content consisting of 50,078 Web pages for training, and 10,024 for testing, with promising results(Dumais and Chen, 2000). Both of them use hierarchies which are manually constructed. Such hierarchies are costly human intervention, since the number of categories and the size of the target corpora are usually very large. Further, manually constructed hierarchies are very general in order to meet the needs of a large number of forthcoming accessible source of text data, and sometimes constructed by relying on human intuition. Therefore, it is difficult to keep consistency, and thus, problematic for classifying text automatically.

In this paper, we address the problem dealing with a large collection of data, and propose a method to generate category hierarchy for text classification. Our method uses two well-known techniques, partitioning clustering method called k-means and a $loss\ function$ to create hierarchical structure. k-means partitions a set of given categories into k clusters, locally minimizing the average squared distance between the data points and the cluster centers. The algorithm involves iterating through the data that the system is permitted to classify during each iteration and constructs category hierarchy. To select the proper number of k during each iteration, we use a $loss\ function$ which measures the degree of our disappointment in any differences between the true distribution over inputs and the learner's prediction. Another focus of this paper is whether or not a large collection of data, the 1996 Reuters corpus helps to generate a category hierarchy which is used to classify documents.

The rest of the paper is organized as follows. The next section presents a brief review the earlier work. We then explain the basic framework for constructing category hierarchy, and describe hierarchical classification. Finally, we report some experiments using the 1996 Reuters corpus with a discussion of evaluation.

2 Related Work

Automatically generating hierarchies is not a new goal for NLP and their application systems, and there have been several attempts to create various types of hierarchies(Koller and Sahami, 1997), (Nevill-Manning et al., 1999), (Sanderson and Croft, 1999). One attempt is Crouch(Crouch, 1988), which automatically generates thesauri. Cutting et al. proposed a method called Scatter/Gather in which clustering is used to create document hierarchies(Cutting et al., 1992). Lawrie et al. proposed a method to create domain specific hierarchies that can be used for browsing a document set and locating relevant documents(Lawrie and Croft, 2000).

At about the same time, several researchers have investigated the use of automatically generating hierarchies for a particular application, text classification. Iwayama et al. presented a probabilistic clustering algorithm called Hierarchical Bayesian Clustering(HBC) to construct a set of clusters for text classification(Iwayama and Tokunaga, 1995). The searching platform they focused on is the probabilistic model of text categorisation that searches the most likely clusters to which an unseen document is classified. They tested their method using two data sets: Japanese dictionary data called 'Gendai yogo no kisotisiki' which contains 18,476 word entries, and a collection of English news stories from the Wall Street Journal which consists of 12,380 articles. The HBC model showed 2∼3% improvements in breakeven point over the non-hierarchical model.

Weigend et al. proposed a method to generate hierarchies using a probabilistic approach(Weigend et al., 1999). They used an exploratory cluster analysis to create hierarchies, and this was then verified by human assignments. They used the Reuters-22173 and defined two-level categories: 5 top-level categories (agriculture, energy, foreign exchange, metals and miscellaneous category) called meta-topic, and other category groups assigned to its meta-topic. Their method is based on a probabilistic approach that frames the learning problem as one of function approximation for the posterior probability of the topic vector given the input vector. They used a neural net architecture and explored several input representations. Information from each level of the hierarchy is combined in a multiplicative fashion, so no hard decision have to be made except at the leaf nodes. They found a 5% advantage in average precision for the hierarchical representation when using words.

All of these mentioned above perform well, while the collection they tested is small compared with many realistic applications. In this paper, we investigate that a large collection of data helps to generate a hierarchy, i.e. it is statistically significant better than the results which utilize hierarchical structure by hand, that has not previously been explored in the context of hierarchical classification except for the improvements of hierarchical model over the flat model.

3 Generating Hierarchical Structure

3.1 Document Representation

To generate hierarchies, we need to address the question of how to represent texts(Cutting et al., 1992), (Lawrie and Croft, 2000). The total number of words we focus on is too large and it is computationally very expensive.

We use two statistical techniques to reduce the number of inputs. The first is to use $category\ vector$ instead of $document\ vector$. The number of input vectors is not the number of the training documents but equals to the number of different categories. This allows to make the large collection of data more manageable. The second is a well-known technique, i.e. mutual information measure between a word and a category. We use it as the value in each dimension of the vector(Cover and Thomas, 1991). More formally, each category in the training set is represented using a vector of weighted words. We call it $category\ vector$. Category vectors are used for representing as points in Euclidean space in k-means clustering algorithm. Let c_j be one of the categories $c_1, \cdots, c_m$, and a vector assigned to c_j be $(c_{1j}, c_{2j}, \cdots, c_{nj})$. The mutual information $MI(W, Cat)$ between a word W, and a category Cat is defined as:

$$MI(W, Cat) \\ = \sum_{W \in \{w, \bar{w}\}} \sum_{Cat \in \{c, \bar{c}\}} P(W, Cat) \log \frac{P(W, Cat)}{P(W)P(Cat)} \quad (1)$$

Each c_{ij} ($1 \leq i \leq n$) is the value of mutual information between w_i and c_j. We select the 1,000 words with the largest mutual information for each category.

3.2 Clustering

Clustering has long been used to group data with many applications(Jain and Dubes, 1988). We use a simple clustering technique, k-means to group categories and construct a category hierarchy(Duda and Hart, 1973). k-means is based on iterative relocation that partitions a dataset into k clusters. The algorithm keeps track of the centroids, i.e. seed points, of the subsets, and proceeds in iterations. In each iteration, the following is performed: (i) for each point x, find the seed point which is closest to x. Associate x with this seed point, (ii) re-estimate each seed point locations by taking the center of mass of points associated with it. Before the first iteration the seed points are initialized to random values. However, a bad choice of initial centers can have a great impact on performance, since k-means is fully deterministic, given the starting seed points. We note that by utilizing hierarchical structure, the classification problem can be decomposed into a set of smaller problems corresponding to hierarchical splits in the tree. This indicates that one first learns rough distinctions among classes at the top level, then lower level distinctions are learned only within the appropriate top level of the tree, and lead to more specialized classifiers. We thus selected the top k frequent categories as initial seed points. Figure 1 illustrates a sample hierarchy obtained by k-means. The input is a set of category vectors. Seed points assigned to each cluster are underlined in Figure 1.

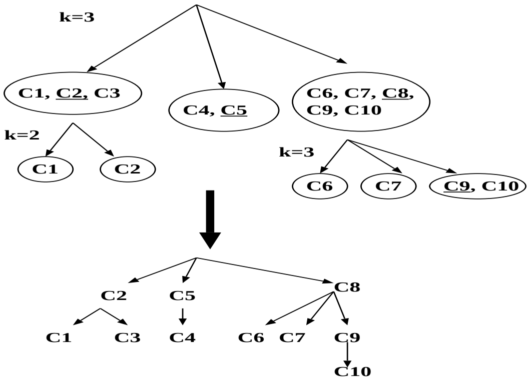

Figure 1: Hierarchical structure obtained by k-means

In general, the number of k is not given beforehand. We thus use a $loss\ function$ which is derived from Naive Bayes(NB) classifiers to evaluate the goodness of k.

3.3 NB

Naive Bayes(NB) probabilistic classifiers are commonly studied in machine learning(Mitchell, 1996). The basic idea in NB approaches is to use the joint probabilities of words and categories to estimate the probabilities of categories given a document. The NB assumption is that all the words in a text are conditionally independent given the value of a classification variable. There are several versions of the NB classifiers. Recent studies on a Naive Bayes classifier which is proposed by McCallum et al. reported high performance over some other commonly used versions of NB on several data collections(McCallum, 1999). We use the model of NB by McCallum et al. which is shown in formula (2).

$$P(c_j \mid d_i, \hat{\theta}) \quad = \quad \frac{P(c_j \mid \hat{\theta}) \Pi_{k=1}^{|d_i|} P(w_{d_{ik}} \mid c_j, \hat{\theta})}{\sum_{r=1}^{|C|} P(c_r \mid \hat{\theta}) \Pi_{k=1}^{|d_i|} P(w_{d_{ik}} \mid c_r, \hat{\theta})}$$

$where$

$$\begin{aligned} \hat{\theta}_{tj} &\equiv P(w_t \mid c_j, \hat{\theta}) \\ &= \frac{1 + \sum_{i=1}^{|D|} N(w_t, d_i) P(c_j \mid d_i)}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N(w_s, d_i) P(c_j \mid d_i)} \\ \hat{\theta}_{0j} &\equiv P(c_j \mid \hat{\theta}) = \sum_{i=1}^{|D|} P(c_j \mid d_i)/|D| \end{aligned} \tag{2}$$

$|V|$ refers to the size of vocabulary, $|D|$ denotes the number of labeled training documents, and $|C|$ shows the number of categories. $|d_i|$ denotes document length. $w_{d_{ik}}$ is the word in position k of document d_i, where the subscript of w, d_{ik} indicates an index into the vocabulary. $N(w_t, d_i)$ denotes the number of times word w_t occurs in document d_i, and $P(c_j \mid d_i)$ is defined by $P(c_j \mid d_i)$ $\in \{0,1\}$.

There are several strategies for assigning categories to a document based on the probability $P(c_j \mid d_i, \hat{\theta})$ such as k-per-doc strategy (Field, 1975), $probability\ threshold$ and $proportional\ assignment$ strategies(Lewis, 1992). We use probability threshold(PT) strategy where each document is assigned to the categories above a threshold θ[1]. The threshold θ can be set to control precision and recall. Increasing θ, results in fewer test items meeting the criterion, and this usually increases precision but decreases recall. Conversely, decreasing θ typically decreases precision but increases recall. In a flat non-hierarchical model, we chose θ for each category, so as to optimize performance on the F measure on a training samples and development test samples. In a manual and automatic construction of hierarchy, we chose θ at each level of a hierarchy using training samples and development test samples.

3.4 Estimating Error Reduction

Let $P(y \mid x)$ be an unknown conditional distribution over inputs, x, and output classes, $y \in \{y_1, y_2, \cdots, y_n\}$, and let $P(x)$ be the marginal 'input' distribution. The learner is given a labeled training set D, and estimates a classification function that, given an input x, produces an estimated output distribution $\hat{P}_D(y \mid x)$. The expected error of the learner can be defined as follows:

$$E_{\hat{P}_D} \quad = \quad \int_x L(P(y \mid x), \hat{P}_D(y \mid x)) P(x) \tag{3}$$

where L is some loss function that measures the degree of our disappointment in any differences between the true

[1] We tested these three assignment strategies in the experiment, and obtained a better result with probability threshold than with other strategies.

distribution, $P(y \mid x)$ and the learner's prediction, $\hat{P}_D(y \mid x)$. A log loss which is defined as follows:

$$L \quad = \quad \sum_{y \in Y} P(y \mid x) \log(\hat{P}_D(y \mid x)) \qquad (4)$$

Suppose that we chose the optimal number of k in the k-means algorithm. Let D_k ($2 \leq k \leq n$) be one of the result obtained by k-means algorithm, and be a set of seed points(categories) labeled training samples. The learner aims to select the result of D_i, such that the learner trained on the set D_i has lower error rate than any other sets.

$$\forall k \;\; E_{\hat{P}_{D_i}} < E_{\hat{P}_{D_k}} \qquad (5)$$

We defined a loss function as follows:

$$\hat{E}_{\hat{P}_{D_i}} = -\frac{1}{|Y_k|} \frac{1}{|X|} \sum_{x \in X} \sum_{y \in Y_k} P(y \mid x) \log(\hat{P}_{D_i}(y \mid x)) \quad (6)$$

Y_k in formula (6) denotes a set of seed points(categories) of D_k. We note that the true output distribution $P(y \mid x)$ in formula (6) is unknown for each sample x. Roy et al.(Roy and McCallum, 2001) proposed a method of $active\ learning$ that directly optimizes expected future error by log-loss, using the entropy of the posterior class distribution on a sample of the unlabeled examples. We applied their technique to estimate it using the current learner. More precisely, from the development training samples D, a different training set is created. The learner then creates a new classifier from the set. This procedure is repeated m times, and the final class posterior for an instance is taken to be the average of the class posteriori for each of the classifiers.

3.5 Generating Category Hierarchy

The algorithm for generating category hierarchy is as follows:

1. Create category vectors from the given training samples.
2. Apply k-means up to n-1 times ($2 \leq k \leq n$), where n is the number of different categories.
3. Apply a loss function (6) to each result.
4. Select the i-th result using formula (5), i.e. the result such that the learner trained on the i-th set has lower error rate than any other results.
5. Assign every seed point(category) of the clusters in the i-th result to each node of the tree.

For each cluster of sub-branches, eliminates the seed point, and the procedure 2 $\sim$ 5 is repeated, i.e. run a local k-means for each cluster of children, until the number of categories in each cluster is less than two.

4 Hierarchical Classification

Like Dumais's approach(Dumais and Chen, 2000), we classify test data using the hierarchy. We select the 1,000 features with the largest mutual information for each category, and use them for testing. The selected features are used as input to the NB classifiers.

We employ the hierarchy by learning separate classifiers at each internal node of the tree. Then using these classifiers, we assign categories to each test sample using probability threshold strategy where each sample is assigned to categories above a threshold θ. The process is repeated by greedily selecting sub-branches until it reaches a leaf.

5 Evaluation

5.1 Data and Evaluation Methodology

We compare automatically created hierarchy with flat and manually constructed hierarchy with respect to classification accuracy. We further evaluate the generated category hierarchy from two perspectives: we examine (i) whether or not the number of categories effects to construct a category hierarchy, and (ii) whether or not a large collection of data helps to generate a category hierarchy.

The data we used is the 1996 Reuters corpus which is available lately(Reuters, 2000). The corpus from 20th Aug., 1996 to 19th Aug., 1997 consists of 806,791 documents. These documents are organized into 126 topical categories with a fifth level hierarchy. After eliminating unlabeled documents, we divide these documents into four sets. Table 1 illustrates each data which we used in each model, i.e. a flat non-hierarchical model, manually constructed hierarchy, and automatically created hierarchy. The same notation of (X) in Table 1 denotes a pair of training and test data. For example, '(F1) Training data' shows that 145,919 samples are used for training NB classifiers, and '(F1) Test data' illustrates that 290,665 samples are used for classification. We selected 102 categories which have at least one document in each data.

We obtained a vocabulary of 320,935 unique words after eliminating words which occur only once, stemming by a part-of-speech tagger(Schmid, 1995), and stop word removal. The number of categories per document is 3.21 on average. For both of the hierarchical and non-hierarchical cases, we select the 1,000 features with the largest MI for each of the 102 categories, and create $category\ vector$.

Like Roy et al's method, we use $bagging$ to reduce variance of the true output distribution $P(y \mid x)$. From

Table 1: Data sets used in each method

	Training samples	Dev. training samples	Dev. test samples	Test samples
# of samples	(145,919 samples)	(300,000 samples)	(60,021 samples)	(290,665 samples)
Date	'96/08/20-'96/10/30	'96/10/30-'97/03/19	'97/03/19-'97/04/01	'97/04/01-'97/08/19
Flat	(F1) Training data	(F2) Training data for estimating PT	(F2) Test data for estimating PT	(F1) Test data
Manual	(M1) Training data	(M2) Training data for estimating PT at each hierarchical level	(M2) Test data for estimating PT at each hierarchical level	(M1) Test data
Automatic	(A1) Training data	(A2) Training data for estimating PT at each hierarchical level (A3) Training data for estimating the true output distribution	(A2) Test data for estimating PT at each hierarchical level (A3) Test data for estimating the true output distribution	(A1) Test data

our original development training set, 300,000 documents, a different training set which consists of 200,000 documents is created by random sampling. The learner then creates a new NB classifier from this sample. This procedure is repeated 10 times, and the final class posterior for an instance is taken to be the average of the class posteriors for each of the classifiers.

For evaluating the effectiveness of category assignments, we use the standard recall, precision, and F-score. Recall is defined to be the ratio of correct assignments by the system divided by the total number of correct assignments. Precision is the ratio of correct assignments by the system divided by the total number of the system's assignments. The F-score which combines recall (r) and precision (p) with an equal weight is F$(r, p) = \frac{2rp}{r+p}$. We use micro-averaging F score where it computes globally over n(all the number of categories) × m (the number of total test documents) binary decisions.

5.2 Results and Discussion

5.2.1 Generating Category Hierarchy

Table 2 shows a top level of the hierarchy which is manually constructed. Table 3 shows a portion of the automatically generating hierarchy which is associated with the categories shown in Table 2. 'x-y-$\cdots$' shows the ID number which is assigned to each node of the tree. For example, **3-5-1** shows that the ID number of the top, second, and third level is **3**, **5**, and **1**, respectively. θ shows a threshold value obtained by the training samples and development test samples.

Tables 2 and 3 indicate that the automatically constructed hierarchy has different properties from manually created hierarchies. When the top level categories of hierarchical structure are equally $discriminating$ properties, they are useful for text classification. In the manual construction of hierarchy(Reuters, 2000), there are 25 categories in the top level, while the result of our method based on corpus statistics shows that the top 4 frequent categories are selected as a discriminative properties, and other categories are sub-categorised into 'Government/social' except for 'Labour issues' and 'Weather'. In the automatically generating hierarchy, 'Economics' → 'Expenditure' → 'Welfare', and 'Economics' → 'Labour issues' are created, while 'Welfare' and 'Labour issues' belong to the top level in the manually constructed hierarchy.

Another interesting feature of our result is that some of the related categories are merged into one cluster, while in the manual hierarchy, they are different locations. Table 4 illustrates the sample result of related categories in the automatically created hierarchy. In Table 4, for example, 'Ec competition/subsidy' is sub-categorised by 'Monopolies/competition'. In a similar way, 'E31', 'E311', 'E143', and 'E132' are classified into 'MCAT', since these categories are related to market news.

5.2.2 Classification

As just described, thresholds for each level of a hierarchy were established on the training samples and development test samples. We then use these thresholds for text classification, i.e. for each level of a hierarchy, if a test sample exceeds the threshold, we assigned the category to the test sample. A test sample can be in zero, one, or more than one categories.

Table 5 shows the result of classification accuracy. 'Flat' and 'Manual' shows the baseline, i.e. the result for all 102 categories are treated as a flat non-hierarchical problem, and the result using manually constructed hierarchy, respectively. 'Automatic' denotes the result of our method. 'miR', 'miP', and 'miF' refers to the micro-averaged recall, precision, and F-score, respectively.

Table 5 shows that the overall F values obtained by our method was 3.9% better than the Flat model, and 3.3% better than the Manual model. Both results are sta-

Table 2: Manually constructed hierarchies

Level	θ	Category node			
Top	0.400	**1** Corporate/industrial	**2** Economics	**3** Government/social	**4** Markets
		5 Crime	**6** Defence	**7** International relations	**8** Disasters
		9 Entertainment	**10** Environment	**11** Fashion	**12** Health
		13 Labour issues	**14** Obituaries	**15** Human interest	**16** Domestic politics
		17 Biographies/people	**18** Religion	**19** Science	**20** Sports
		21 Tourism	**22** War	**23** Elections	**24** Weather
		25 Welfare			

Table 3: Automatically constructed hierarchies

Level	θ	Category node			
Top	0.422	**1** Corporate/industrial	**2 Economics**	**3** Government/social	**4** Markets
Second	0.098	**3-1** Domestic politics	**3-2** International relations	**3-3** War	**3-4** Crime
		3-5 Merchandise trade	**3-6** Sports	**3-7** Defence	**3-8** Disasters
		3-9 Elections	**3-10** Biographies/people	**4-10** Weather	**2-5 Expenditure**
		2-6 Labour issues			
Third	0.992	**3-1-1** Health	**3-5-1** Tourism	**3-8-1** Environment	**3-10-1** Entertainment
		3-10-2 Religion	**3-10-3** Science	**3-10-4** Human interest	**3-10-5** Obituaries
		2-5-1 Welfare			
Fourth	0.999	**3-10-4-1** Fashion			

Table 5: Classification accuracy

Method	miR	miP	miF
Flat	0.753	0.647	0.695
Manual	0.685	0.708	0.701
Automatic	0.807	0.675	0.734

tistically significant using a micro sign test, P-value $\leq$ .01(Yang and Liu, 1999). Somewhat surprisingly, there is no difference between Flat and Manual, since a micro sign test, P-value $>$ 0.05. This shows that manual construction of hierarchy which depends on a corpus is a difficult task. The overall F value of our method is 0.734. Classifying large data with similar categories is a difficult task, so we did not expect to have exceptionally high accuracy like Reuters-21578 (the performance over 0.85 F-score(Yang and Liu, 1999)). Performance on the closed data, i.e. training samples and development test samples in 'Flat', 'Manual', and 'Automatic' was 0.705, 0.720, and 0.782, respectively. Therefore, this is a difficult learning task and generalization to the test set is quite reasonable.

Table 6 and Table 7 illustrates the results at each hierarchical level of manually constructed hierarchies, and our method, respectively. 'Clusters' denotes the number of clusters, and 'Categories' refers to the number of categories at each level. The F-score of 'Manual' for the top level categories is 0.744, and that of our method is 0.919. They outperform the flat model. However, the performance by both 'Manual' and our method monotonically decreases when the depth from the top level to each node is large, and the overall F-score at the lower level of hierarchies is very low. This is because at the lower level more similar or the same features could be used as features within the same top level category. This suggests that we should be able to obtain further advantages in efficiency in the hierarchical approach by reducing the number of features which are not useful discriminators within the lower-level of hierarchies(Koller and Sahami, 1997).

Table 6: Accuracy by hierarchical level(Manual)

Level	Clusters	Categories	miR	miP	miF
Top	25	25	0.753	0.724	0.744
Second	16	63	0.524	0.553	0.543
Third	37	37	0.431	0.600	0.500
Fourth	43	43	0.276	0.103	0.146
Fifth	1	1	0	0	0

Table 7: Accuracy by hierarchical level(Automatic)

Level	Clusters	Categories	miR	miP	miF
Top	4	4	0.988	0.859	0.919
Second	59	59	0.813	0.697	0.750
Third	35	35	0.568	0.126	0.151
Fourth	4	4	0.246	0.102	0.103

Table 4: The sample result of related categories

Node	Category	Node	Category	Node	Category
1-22	Monopolies/competition(C34)	**1-22-1**	Ec competition/subsidy(G157)		
2-7	Defence(GDEF)	**2-7-1**	Defence contracts(C331)		
3	Markets(MCAT)	**3-11**	Output/capacity(E31)		
		3-12	Industrial production(E311)		
		3-13	Retail sales(E143)	**3-13-1**	Housing starts(E61)
		3-14	Wholesale prices(E132)		
4	Economics(ECAT)	**4-2**	Monetary/economic(E12)		
		4-8	Ec monetary/economic(G154)		
4	Economics(ECAT)	**4-6**	Labour issues(GJOB)		
		4-7	Employment/labour(E41)	**4-7-1**	Unemployment(E411)

5.2.3 The Number of Categories v.s. Accuracy

Figure 2 shows the result of classification accuracy using different number of categories, i.e. 10, 50 and 102 categories. Each set of 10 and 50 categories from training samples is created by random sampling. The sampling is repeated 10 times[2]. Each point in Figure 2 is the average performance over 10 sets.

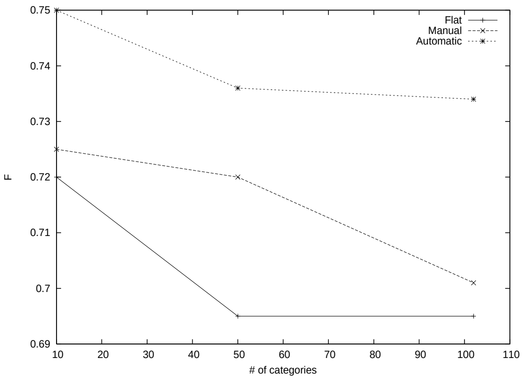

Figure 2: Accuracy v.s. category size

Figure 2 shows that our method outperforms the flat and manually constructed hierarchy at every point in the graph. As can be seen, the grater the number of categories, the more likely it is that a test sample has been incorrectly classified. Specifically, the performance of flat model using 10 categories was 0.720 F-score and that of 50 categories was 0.695. This drop of accuracy indicates that flat model is likely to be difficult to train when there are a large number of classes with a large number of features.

5.2.4 Efficiency of Large Corpora

Figure 3 shows the result of classification accuracy using the different size of training samples, i.e. 10,000, 100,000 and 145,919 samples[3]. Each set of samples is created by random sampling except for the set of 145,919 samples. The sampling process is repeated 10 times. The average accuracy of each result across the 10 sets of samples is reported in Figure 3.

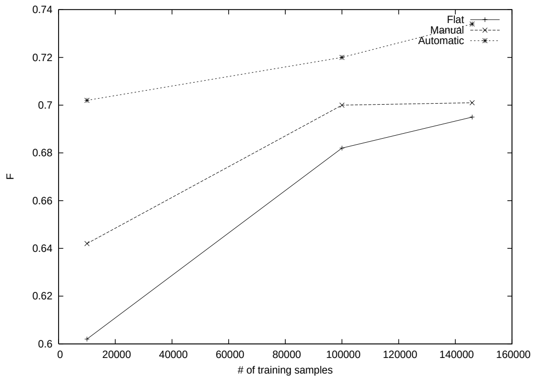

Figure 3: Accuracy v.s. training corpus size

The performance can benefit significantly from much larger training samples. At the number of training samples is 10,000, all methods have poor effectiveness, but it learns rapidly, especially, the results of flat model shows that it is extremely sensitive to the amount of training data. At every point in the graph, the result of our method with cluster-based generalisations outperforms other two methods, especially, the method becomes more attractive with less training samples.

[2] In our method, 10 hierarchies are constructed for each set of categories.

[3] We used the same number of development, development test, and test samples which are shown in Table1 in the experiment.

6 Conclusions

We proposed a method for generating category hierarchy in order to improve text classification performance. We used k-means and a $loss\ function$ which is derived from NB classifiers. We found small advantages in the F-score for automatically generated hierarchy, compared with a baseline flat non-hierarchy and that of manually constructed hierarchy from large training samples. We have also shown that our method can benefit significantly from less training samples. Future work includes (i) extracting features which discriminate between categories within the same cluster with low F-score, (ii) using other machine learning techniques to obtain further advantages in efficiency in dealing with a large collection of data, (iii) comparing the method with other techniques such as hierarchical agglomerative clustering and 'X-means'(Pelleg and Moore, 2000), and (iv) developing evaluation method between manual and automatic construction of hierarchies to learn more about the strengths and weaknesses of the two methods of classifying documents.

Acknowledgments

We would like to thank anonymous reviewers for their valuable comments. We also would like to express many thanks to the Research and Standards Group of Reuters who provided us the corpora.

References

T. Cover and J. Thomas. 1991. Elements of Information Theory. In *Wiley*.

C. Crouch. 1988. A Cluster-based Approach to Thesaurus Construction. In *Proc. of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 309–320.

D. Cutting, D. Karger, J. Pedersen, and J. Tukey. 1992. Scatter/gather: A Cluster-based Approach to Browsing Large Document Collections. In *Proc. of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–329.

T.G. Dietterich. 2000. Ensemble Methods in Machine Learning. In *Proc. of the 1st International Workshop on Multiple Classifier Systems*.

R. Duda and P. Hart. 1973. *Pattern Classification and Scene Analysis*. Wiley.

S. Dumais and H. Chen. 2000. Hierarchical Classification of Web Content. In *Proc. of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 256–263.

B. Field. 1975. Towards Automatic Indexing: Automatic Assignment of Controlled Language Indexing and Classification from Free Indexing. *Journal of Documentation*, pages 246–265.

M. Iwayama and T. Tokunaga. 1995. Cluster-Based Text Categorization: A Comparison of Category Search Strategies. In *Proc. of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 273–280.

A.K. Jain and R.C. Dubes. 1988. *Algorithms for Clustering Data*. Englewood Cliffs N.J. Prentice Hall.

D. Koller and M. Sahami. 1997. Hierarchically Classifying Documents using Very Few Words. In *Proc. of the 14th International Conference on Machine Learning*, pages 170–178.

D. Lawrie and W.B. Croft. 2000. Discovering and Comparing Hierarchies. In *Proc. of RIAO 2000 Conference*, pages 314–330.

D.D. Lewis. 1992. An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. In *Proc. of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 37–50.

A.K. McCallum. 1999. Multi-Label Text Classification with a Mixture Model Trained by EM. In *Revised version of paper appearing in AAAI'99 Workshop on Text Learning*.

T. Mitchell. 1996. *Machine Learning*. McGraw Hill.

Nevill-Manning, C.I. Witten, and G. Paynter. 1999. Lexically-Generated Subject Hierarchies for Browsing Large Collections. *International Journal on digital Libraries*, 2(3):111–123.

D. Pelleg and A. Moore. 2000. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *Proc. of the 17th International Conference on Machine Learning*, pages 725–734.

Reuters. 2000. *Reuters Corpus Volume 1 English Language*. 1996-08-20 to 1997-08-19, release date 2000-11-03, Format version 1, http://www.reuters.com/researchandstandards/corpus/.

N. Roy and A.K. McCallum. 2001. Toward Optimal Active Learning through Sampling Estimation of Error Reduction. In *Proc. of the 18th International Conference on Machine Learning*, pages 441–448.

M. Sanderson and B. Croft. 1999. Deriving Concept Hierarchies from Text. In *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 206–213.

H. Schmid. 1995. Improvements in Part-of-Speech Tagging with an Application to German. In *Proc. of the EACL SIGDAT Workshop*.

A.S. Weigend, E.D. Wiener, and J.O. Pedersen. 1999. Exploiting Hierarchy in Text Categorization. *Information Retrieval*, 1(3):193–216.

Y. Yang and X. Liu. 1999. A Re-Examination of Text Categorization Methods. In *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49.