Correcting Category Errors in Text Classification

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 category annotation errors which deteriorate the overall performance of text classification. We use two techniques. The first is *support vectors* which are extracted from the training samples by a machine learning technique, Support Vector Machines(SVM). The second is a *loss function* which measures the degree of our disappointment in any differences between the true distribution over inputs and the learner's prediction. We apply it to the extracted support vectors, and correct annotation errors. Experimental results with the RWCP and the Reuters 1996 corpora show that our method achieves high precision in detecting and correcting annotation errors. Further, results on text classification improves accuracy.

1 Introduction

A large number of tagged corpora are widely used in corpus-based NLP and their application systems. The performance of these systems greatly depends on the quantity and quality of corpora, since they use some statistics or learning algorithms to train a classifier, given a set of tagged examples. One key difficulty with the use of tagged corpora is that they are error prone, since tagging must often be done by a human. For large corpora it is difficult to keep consistency even if tagging has been done by several experts, and thus, problematic for corpus-based NLP with high accuracy.

There are at least two strategies for automatically detecting errors in corpora. One is to use weight which is assigned to each training example by some learning techniques. Abney et al. proposed a method to improve data quality by using boosting. They applied their technique to part-of-speech tagging and prepositional phrase attachment(Abney et al., 1999). Nakagawa et al. used SVM to identify part-of-speech annotation errors in corpora(Nakagawa and Matsumoto, 2002). Both methods utilize the fact that the training examples with larger values of weights are difficult to classify. Such training example tends to be an outlier, and be an annotation error. Abney et al. conducted error detection in the Penn Treebank WSJ corpus by extracting examples with a large weight. Nakagawa et al. tested their method using three different real-world datasets: the Penn Treebank WSJ corpus, the RWCP corpus and the Kyoto University Corpus with high precision.

The other is probabilistic approaches. Eskin proposed a method for detecting part-of-speech annotation errors in a corpus using an anomaly detection technique(Eskin, 2000). The technique is the process of determining when an element of data is an outlier. Eskin used a 'mixture model' with two probability distributions: a majority distribution and an anomalous distribution. For each element, he measured the likelihood of the distribution under both cases. The element is detected as an error if the likelihood in the anomalous distribution is sufficiently large. All of these mentioned above perform well, while they have not applied their methods to correcting annotation errors.

This paper proposes a method to detect and correct category annotation errors which deteriorate the overall performance of text classification. We use two techniques. The first is support vectors which are extracted from the training samples by SVM(Vapnik, 1995). Training SVM is to find the optimal hyperplane which consists of support vectors, and only the support vectors affect the performance. Thus, if some training sample deteriorates the overall performance of text classification because of an outlier, we can assume that the sample is a support vector. The second is a loss function which measures the degree of our disappointment in any differences between the true distribution over inputs and the learner's prediction. We apply it to the extracted support vectors, and

correct annotation errors.

2 Classifiers

We use SVM and NB, mainly for the following reasons. First, we can obtain only the samples in given training data which matter the overall performance with SVM, since training SVM is to find the optimal hyperplane which consists of support vectors. Second, NB classifier is a calibrated posterior probability to enable post-processing, i.e. correction of annotation errors. Third, NB is based on the assumption of word independence in a text, which makes the computation of it far more efficient than SVM.

2.1 SVM

SVM is introduced by Vapnik(Vapnik, 1995) for solving two-class pattern recognition problems. Given training samples $L = \{(\mathbf{x}_1,y_1), (\mathbf{x}_2,y_2), \cdots, (\mathbf{x}_l,y_l)\}$, $(\mathbf{x}_i \in R^n,\ y_i \in \{+1,\ -1\})$, SVM finds a hyperplane that *best* separates a set of positive examples from a set of negative examples. The optimal hyperplane to separate them is found by solving the following optimization problem:

$$\text{minimize} : \tau(\alpha) = -\sum_{i=1}^{n} \alpha_i + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

$$\text{subject to} : \sum_{i=1}^{n} y_i \alpha_i = 0 \qquad \forall_{i=1}^{n}\ 0 \le \alpha_i \le C$$

where C is a parameter that controls the training errors, and becomes upper bound of α_i. Support vectors are those training examples $\mathbf{x}_i$ with $\alpha_i > 0$ at the solution. It is known that we can obtain the same decision function even if we remove all training examples except for support vectors. We use a weight α_i to extract outliers.

SVM is basically introduced for solving binary classification, while text classification is a multi-class, multi-label classification problem. Several methods which were intended for multi-class, multi-label data have been proposed(Weston and Watkins, 1998). We use *One-against-the-Rest* version of the SVM model in the work.

2.2 NB

We use NB for two tasks. The first task is to assign categories to the extracted training samples(support vectors) with SVM, and extract error candidates from the support vectors. The second is to calculate the estimated error for each candidate using a loss function.

The basic idea of NB is to use the joint probabilities of words and categories to estimate the probabilities of categories given a document. There are several versions of the NB classifiers. Recent studies 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 (1).

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

where

$$\hat{\theta}_{tj} \equiv P(w_t \mid c_j, \hat{\theta}) = \frac{1 + \sum_{i=1}^{|D|} N(w_t, d_i) P(c_j \mid d_i)}{|V| + \sum_{s=1}^{|V|}\sum_{i=1}^{|D|} N(w_s, d_i) P(c_j \mid d_i)}$$

$$\hat{\theta}_{0j} \equiv P(c_j \mid \hat{\theta}) = \sum_{i=1}^{|D|} P(c_j \mid d_i)/|D| \qquad (1)$$

$|V|$ 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 $P(c_j|d_i)$ is defined by $P(c_j|d_i) \in \{0,1\}$.

3 Correcting Annotation Errors

Roy et al. 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(Roy and McCallum, 2001). We applied their technique to detect and correct category annotation errors. Figure 1 illustrates an overview of the system design. It consists of three steps: extracting error candidates, estimating error reduction and correcting annotation errors. These steps are repeated for each category given a labeled training data.

3.1 Extracting Error Candidates

Let D^* be training data consisting of n samples. Each sample $\mathbf{x}_i$ is given a set of label Y_i, i.e. multiple categories. $\mathbf{x}_i^* \in \{\mathbf{x}_1^*, \mathbf{x}_2^*, \cdots, \mathbf{x}_m^*\}$ be support vectors with a weight greater than zero, i.e. $\mathbf{x}_i^*$ for being a positive sample of the given label $y_\alpha \in Y_i$ by training a set of D^*.

We remove $\sum_{k=1}^{m} \mathbf{x}_k^*$ from the training samples D^*. The resulting sample D^{**} is used for training NB, leading to a classification model. This classification model is tested on each support vector, $\mathbf{x}_i^*$ and assigns a set of label, Y_i^*. If

$y_\alpha (\in Y_i)$ is not an element of Y_i^*, we declare $\mathbf{x}_i^*$ an error candidate.

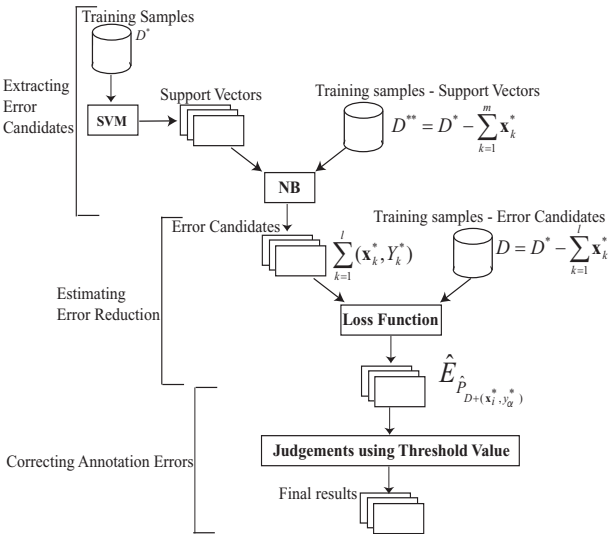

Figure 1: System overview

3.2 Estimating Error Reduction

The optimal active learner is one that asks for labels on the examples that, once incorporated into training, will result in the lowest expected error on the test set. 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 $\hat{P}_D(y|x)$. The expected error of the learner can be defined as follows:

$$E_{\hat{P}_D} = \int_x L(P(y \mid x), \hat{P}_D(y \mid x)) P(x) \quad (2)$$

where L is some loss function that measures the degree of our disappointment in any differences between the true distribution, $P(y \mid x)$ and the learner's prediction, $\hat{P}_D(y|x)$. A log loss which is defined as follows:

$$L = \sum_{y \in Y} P(y \mid x) \log(\hat{P}_D(y \mid x)) \quad (3)$$

The active learning aims to select x', such that when the sample is given label y' and added to the training set, the learner trained on the resulting set $(D + (x', y'))$ has lower error rate than any other x.

$$\forall (x, y) \quad E_{\hat{P}_{D+(x',y')}} < E_{\hat{P}_{D+(x,y)}} \quad (4)$$

Roy et al. defined a loss function as follows:

$$\hat{E}_{\hat{P}_{D+(x',y')}} = -\frac{1}{\mid X \mid} \sum_{x \in X} \sum_{y \in Y} P(y \mid x) \log(\hat{P}_{D+(x',y')}(y \mid x)) \quad (5)$$

We note that the true output distribution $P(y|x)$ in formula (5) is unknown for each sample x. Roy et al. used *bagging* to estimate it. More precisely, from the training samples D, a different training set is created. The learner then creates a new classifier from the sample, this procedure is repeated m times, and the final class posterior for an instance is taken to be the unweighted average of the class posteriori for each of the classifiers.

We recall that detecting error is done by determining whether the label of an error candidate $\mathbf{x}_i^*$ is y_α^* $(\in Y_i^*)$ or not. Here, Y_i^* refers to a set of the resulting category label of $\mathbf{x}_i^*$ which is estimated by the NB classifier given an input $D^{**} = D^*$ - $\sum_{k=1}^{m} \mathbf{x}_k^*$. We then use a loss function in formula (5). Specifically, $P\ (y|x)$ denotes the true distribution, and $\hat{P}_{D+(x',y')}\ (y|x)$ shows the learner's prediction. D denotes the training samples D^* except for the error candidates $\sum_{k=1}^{l} \mathbf{x}_k^*$. Here, l is the number of error candidates. (x',y') in formula (5) refers to $(\mathbf{x}_i^*, y_\alpha^*)$, $y_\alpha^* \in Y_i^*$. X is a set of test samples, and $\mid X \mid$ denotes the number of test samples. Y shows a set of all categories. For each $y_\alpha^* \in Y_i^*$, if the value of $\hat{E}_{\hat{P}_{D+(\mathbf{x}_i^*, y_\alpha^*)}}$ in formula (5) is sufficiently small, the learner's prediction is close to the true distribution. Like Roy et al's method, we use bagging to reduce variance of the true output distribution $P(y|x)$.

3.3 Correcting Annotation Errors

We use formula (5) for each error candidate $\mathbf{x}_i^*$ in order to determine the label assigned to $\mathbf{x}_i^*$ is either y_α, or $y_\alpha^* \in Y_i^*$. More formally, let D^* be training samples, and $\mathbf{x}_i^*$ $(1 \leq i \leq l)$ be an error candidate. For each y_α^* of $\mathbf{x}_i^*$, we calculate $\hat{E}_{\hat{P}_{D+(\mathbf{x}_i^*, y_\alpha^*)}}$, where D refers to D^* - $\sum_{k=1}^{l} \mathbf{x}_k^*$. We pick up $\mathbf{x}_i^*$ whose loss value is smaller than a certain threshold value, θ. If the label of the selected $\mathbf{x}_i^*$ is y_α^*, we declare the label annotated by humans, y_α an error, and its true label is y_α^*. Otherwise, the label of $\mathbf{x}_i^*$ is y_α.

4 Experiments

We tested detecting and correcting performance. Then, we applied the correction results

to text classification. We use two corpora: the RWCP and the 1996 Reuters(RCV1) data. In the following experiments, we use linear SVM and the upper bound value C is set to 1. Performance is governed by two parameters, the weight α_i assigned by SVM and loss value θ obtained by formula (5). We thus conducted experiments for various values of α_i and θ.

4.1 The RWCP Corpus

The RWCP corpus(Toyoura et al., 1996) consists of 30,207 documents taken from the Mainichi Shimbun Newspaper in Japanese published in 1994(Mainichi, 1995). We use ten categories that appeared most often in the corpus. We select 18,841 documents, each of which has one category to examine a single label classification problem. We divide these documents into four sets. Table 1 illustrates each set.
'Training samples' in Table 1 which consists of three sets denotes the samples for detecting and correcting annotation errors. More precisely, the first fold is used for detecting and correcting annotation errors, and the second(Dev. training) and the third folds(Dev. test) are used to estimate the true output distribution $P(y|x)$[1]. This process is repeated three times so that each fold serves as the source of the detection and correction data. 'Test samples' refers to the test samples which are used for text classification. We obtained a vocabulary of 62,709 unique words after stemming by a part-of-speech tagger, Chasen(Matsumoto et al., 1997).

4.1.1 Detection and Correction

Table 2 shows detecting(correcting) performance. 'Sv.' denotes the total number of support vectors, and 'Ec.' denotes the total number of extracted error candidates across the three folds. Precision of detection(correction) is the ratio of correct assignments of detection(correction) by the system divided by the total number of the system's assignments of detection(correction). The results are examined by hand whether the detected and corrected errors are true errors or not. The evaluation is made by two humans. The classification is determined to be correct if two human judges agree. Precision of Table 2 shows the global accuracy across the three folds. In Table 2, for example, 0.3 of α_i value refers to $0.3 \leq \alpha_i \leq 1$, and 0.20 of θ stands for $\theta \leq 0.20$. For each value of α_i, we tested different threshold values, $\theta (0 \leq \theta \leq 0.5)$[2]. Each value of θ shows the best result among them. The best precision score for detection was 0.820, and correction was 0.760.

We expect that no errors are detected by repeating corpus error correction and manual correction of the mislabeled samples by the system. Figure 2 illustrates the result with $0.1 \leq \alpha_i \leq 1$ and $\theta \leq 0.14$. 'Corrected by the system' denotes the number of samples which are corrected by the system. 'Corrected by a human' refers to the number of samples which are manual correction of the mislabeled samples by the system. We can see that the number of corrected annotation errors decreases rapidly, and no errors are detected in the ninth round.

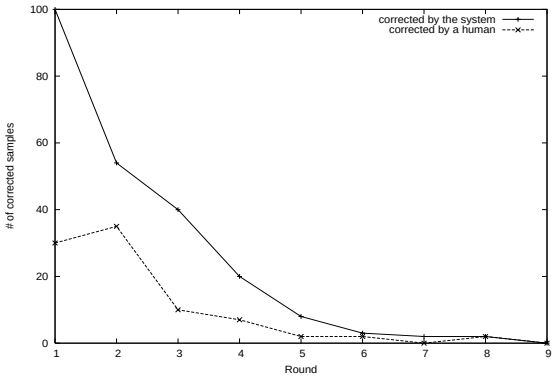

Figure 2: # of corrected errors v.s. round

4.1.2 Text Classification

We applied the best result of correcting annotation errors, i.e. $0 \leq \alpha_i \leq 1$, and $\theta \leq 0.19$ to text classification. 'NB' and 'SVM' in Table 3 denotes the text classification result obtained by NB, and SVM classifiers, respectively. The result of NB shows that we use *k-per-doc* strategy, where $k = 1$(Field, 1975). 'Baseline' refers to the result using all the training samples, i.e. the first three folds. 'Detecting and Correcting' shows the result of our method. The overall F values obtained by our method with NB was 2.3% better than the baseline, and that of SVM was 1.7% better than the baseline. Both results are statistically significant using a micro sign test, P-value ≤ 0.01(Yang and Liu, 1999).

4.2 The Reuters 1996 Corpus

The 1996 Reuters(Reuters, 2000) corpus consists of 806,791 documents from 20th Aug. 1996

[1]For *bagging*, we split Dev. training samples into 5 sets, and create a new classifier from each set. This procedure is repeated 10 times.

[2]We set the upper value of θ to 0.5, since the smaller value of θ is, the more the learner's prediction is close to the true distribution.

Table 1: Data set used in the experiment(RWCP)

	Training samples			Test samples
# of samples	4,448 samples	4,448 samples	4,447 samples	5,498 samples
Date	'94/01/01 - '94/04/18	'94/04/18 - '94/07/18	'94/07/18 - '94/10/30	'94/10/30 - '94/12/31

Table 2: Detecting and correcting accuracy(RWCP)

α_i	Sv.	Ec.	*Loss value* θ	Detecting precision	*Loss value* θ	Correcting precision
1.0	22	8	0.50	0.750(6/8)	0.50	0.750(6/8)
0.9	34	20	0.50	0.750(15/20)	0.50	0.700(14/20)
0.8	38	23	0.20	0.783(18/23)	0.20	0.652(15/23)
0.7	44	24	0.20	0.783(18/23)	0.20	0.652(15/23)
0.6	54	28	0.20	0.800(20/25)	0.20	0.720(18/25)
0.5	71	31	0.30	0.800(24/30)	0.30	0.733(22/30)
0.4	95	42	0.32	0.795(31/39)	0.32	0.718(28/39)
0.3	160	78	0.20	0.810(51/63)	0.20	0.730(46/63)
0.2	306	108	0.20	0.795(58/73)	0.20	0.740(54/73)
0.1	849	297	0.14	**0.820**(82/100)	0.21	**0.760**(76/100)
0	6,435	2,058	0.19	0.700(335/478)	0.19	0.686(328/478)

Table 3: Classification Accuracy(RWCP)

NB classifiers			
Training samples	Recall	Precision	F
Baseline	0.648	0.656	0.652
Detection and correction	0.681	0.670	0.675

SVM classifiers			
Training samples	Recall	Precision	F
Baseline	0.682	0.692	0.687
Detection and correction	0.696	0.713	0.704

to 19th Aug. 1997. These documents are organized into 126 categories with a four level hierarchy. The number of categories in each level is 25 top, 33 second, 43 third, and 1 fourth level, respectively. After eliminating unlabeled documents, we divide these documents into four sets. Table 4 illustrates each set. We use 102 categories which have at least one document in each set. 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.

4.2.1 Detection and Correction

The results are shown in Table 5. Like the RWCP corpus, for each value of α_i, we tested different threshold values, $\theta (0 \leq \theta \leq 0.5)$. Each value in Table 5 shows the best results among them. The best precision for detection was 0.819 and for correction was 0.754. The results are similar to the result using the RWCP corpus, as the best performance of detection was 0.820, and that of correction was 0.760. This shows that the method works well even for multi-label data. In the 1996 Reuters corpus, 2,538 out of 150,000 training samples were error-prone(1,69%). This ratio indicates that nearly 14,000 samples with error-prone would be included in one year corpus, 806,791 samples. Detecting these annotation errors by humans is time-consuming. Our approach detects errors with a high precision(0.819). Thus, even if we annotate only these samples, we can avoid costly human intervention.

Table 6 illustrates the detecting and correcting samples. In the examples 3, 5, and 6, our method mislabeled 'Corporate(CCAT)' to these samples. 519 out of 2,862 samples are incorrectly detected, and 705 out of 2,862 are mislabeled with $0 \leq \alpha_i \leq 1$ and $\theta \leq 0.06$. Of these, 70∼82% of the samples were labeled 'CCAT' incorrectly. The result is quite reasonable because 'CCAT' is assigned to the almost half of the training samples, and thus the category is very general and related to other categories. Yang has shown that the word-category association measures such as a χ^2 statistic and *information gain* criterion are effective for discriminating among categories(Yang and Liu, 1999). This is definitely worth trying with our method.

4.2.2 Text Classification

Table 7 shows the results obtained by using the best result of correction. The result of NB shows that we use *k-per-doc*, where $k = 3$. Ta-

Table 4: Data set used in the experiment(Reuters1996)

	Training samples			Test samples
# of samples	50,000 samples	50,000 samples	50,000 samples	646,605 samples
Date	'96/08/20 - '96/09/13	'96/09/13 - '96/10/08	'96/10/08 - '96/10/30	'96/10/30 - '97/08/19

Table 5: Detecting and correcting accuracy(Reuters1996)

α_i	Sv.	Ec.	*Loss value* θ	Detecting precision	*Loss value* θ	Correcting precision
1.0	6,621	1,980	0.44	0.709(397/560)	0.44	0.657(368/560)
0.9	9,246	2,853	0.34	0.738(549/744)	0.34	0.659(490/744)
0.8	9,918	2,968	0.41	0.740(562/759)	0.41	0.671(509/759)
0.7	10,584	3,024	0.40	0.756(567/750)	0.40	0.681(511/750)
0.6	11,220	3,066	0.41	0.762(608/789)	0.40	0.659(526/798)
0.5	12,198	3,328	0.42	0.777(623/802)	0.42	0.704(565/802)
0.4	13,629	3,487	0.35	0.790(1,122/1,428)	0.34	0.711(1,009/1,420)
0.3	16,215	4,264	0.38	0.806(1,646/2,058)	0.37	0.727(1,485/2,042)
0.2	21,792	4,437	0.15	**0.819**(1,062/1,296)	0.15	0.736(954/1,296)
0.1	39,666	10,398	0.18	0.812(1,314/1,618)	0.18	0.723(1,170/1,618)
0	171,456	45,832	0.06	**0.819**(2,343/2,862)	0.06	**0.754**(2,157/2,862)

ble 7 indicates that the result of NB improves 1.7% and that of SVM is 1.1%. Both results are significantly different than the baseline using a micro sign test, P-value ≤ 0.01.

Table 7: Classification Accuracy(Reuters1996)

NB classifiers			
Training samples	Recall	Precision	F
Baseline	0.720	0.674	0.696
Detection and correction	0.738	0.689	0.713

SVM classifiers			
Training samples	Recall	Precision	F
Baseline	0.743	0.744	0.744
Detection and correction	0.759	0.752	0.755

Performance varies widely across categories. Table 8 illustrates three categories with the best improvement and the worst drop of F scores obtained by SVM classifiers. The most significant category was 'Ownership changes', and the F score of our method was 6.4% better than the baseline. Table 8 shows that the accuracy drops when the depth from the top node is large, as the third level categories such as 'C152' and 'E512' belong to 'the drop of F' class. It might be useful to use category hierarchies, i.e. we employ a hierarchy by learning separate classifiers at each internal node of the tree, and then detecting errors to greedily select sub-branches until a leaf is reached(Dumais and Chen, 2000).

The running cost of SVM depends on the number of features and categories. Training time for 50,000 samples(102 categories) was more than 6 days using a standard 3.4 GHz Pentium IV PC with 2 GB of RAM. Efficiency can be improved if we can reduce the number of features without sacrificing accuracy.

Table 8: Accuracy for each category(Reuters1996)

Improvement of F (SVM classifiers)		
Category	Baseline	Correction result
Ownership changes(C18)	0.688	0.752 (**+0.064**)
Performance(C15)	0.845	0.906 (+0.061)
Commodity markets(M14)	0.877	0.920 (+0.043)

Drop of F (SVM classifiers)		
Category	Baseline	Correction result
Comment/Forecasts(C152)	0.716	0.691 (**-0.025**)
Merchandise trade(E512)	0.476	0.453 (-0.023)
Strategy/Plan(C11)	0.356	0.337 (-0.019)

5 Conclusion

The research described in this paper explores the correction of category annotation errors in corpora, based on integrating information from two different classification algorithms: NB and SVM. We found small advantages in the F score for text classification using the RWCP and the 1996 Reuters corpora, compared with a baseline. Future work includes feature reduction and investigation of other learning techniques to obtain further advantages in efficiency in the manipulating large corpora(Zhang et al., 2003).

Acknowledgments

The authors would like to thank anonymous reviewers for their valuable comments. This work was supported by the Grant-in-aid for the JSPS,

Table 6: Detecting and correcting examples(Reuters1996)

Title	Before	After
Correctly detected and corrected errors		
1: Viag shares down after H1 results.	Performance	Equity market
2: USA: New Pan Am expects debut after Labor Day.	Labour	Strategy, Regulation
Correctly detected, but incorrectly corrected errors		
3: RTRS-Australian companies ex-dividend today.	Accounts	Corporate
4: India overseas country funds rates - August 23.	Accounts	Economics
Incorrectly detected and corrected errors		
5: Toronto indices closing - Aug 23.	Equity markets	Corporate
6: Mexico shrs trading electronically, prices delayed.	Equity markets	Corporate
Unsure		
7: Clinton to accept regulation of tobacco as drug.	Regulation/Policy	Domestic politics
8: India overseas country funds rates - August 23.	Performance	Economics

Research Foundation for the Electro-technology of Chubu, and the Kayamori Foundation of Informational Science Advancement.

References

S. Abney, R.E. Schapire, and Y. Singer. 1999. Boosting Applied to Tagging and PP Attachment. In *Proc. of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 38–45.

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.

E. Eskin. 2000. Detecting Errors within a Corpus using Anomaly Detection. In *Proc. of the 6th Applied Natural Language Processing Conference and the 1st Meeting of the North American Chapter of the Association of Computational Linguistics*, pages 148–153.

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

Mainichi. 1995. *CD Mainichi Shimbun 94*. Nichigai Associates Co.

Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Harano, O. Imaichi, and T.Imamura. 1997. *Japanese Morphological Analysis System Chasen Mannual*. NAIST Technical Report NAIST-IS-TR97007.

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. Nakagawa and Y. Matsumoto. 2002. Detecting Errors in Corpora Using Support Vector Machines. In *Proc. of the 19th International Conference on Computational Linguistics*, pages 709–715.

Reuters, 2000. *Reuters Corpus Volume1 English Language, 1996-08-20 to 1997-08-19 Release Date 2000-11-03 Format Version 1*. Reuters.

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.

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

J. Toyoura, T. Tokunaga, H. Isahara, and R. Oka. 1996. Development of a RWC Text Database Tagged with Classification Code(in Japanese). In *NLC96-13, IEICE*, pages 89–96.

V. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer, New York.

J. Weston and C. Watkins. 1998. Multi-Class Support Vector Machines. In *Technical Report CSD-TR-98-04*.

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.

J. Zhang, R. Jin, Y. Yang, and A.G. Hauptmann. 2003. Modified Logistic Regression: An Approximation to SVM and its Applications in Large-Scale Text Categorization. In *Proc. of the 20th International Conference on Machine Learning*, pages 856–863.