A New Pairwise Ensemble Approach for Text Classification

Yan Liu, Jaime Carbonell, Rong Jin

School of Computer Science,
Carnegie Mellon University,
5000 Forbes Ave, Pittsburgh, USA
{yanliu, jgc, rong+}@cs.cmu.edu

Abstract. Text classification, whether by topic or genre, is an important task that contributes to text extraction, retrieval, summarization and question answering. In this paper we present a new pairwise ensemble approach, which uses pairwise Support Vector Machine (SVM) classifiers as base classifiers and "input-dependent latent variable" method for model combination. This new approach better captures the characteristics of genre classification, including its heterogeneous nature. Our experiments on two multi-genre collections and one topic-based classification datasets show that the pairwise ensemble method outperforms both boosting, which has been demonstrated as a powerful ensemble approach, and Error-Correcting Output Codes (ECOC), which applies pairwise-like classifiers for multiclass classification problems.

1 Introduction

Text classification, the problem of assigning documents to predefined categories, is an active research area in both information retrieval and machine learning. It plays an important role in information extraction and summarization, text retrieval, and question-answering. In general, text classification includes topic-based text classification and text genre-based classification. Topic-based text categorization, which is classifying documents according to their topics, has been intensively studied before [24, 26]. However, texts can also be written in many genres, for instance: scientific articles, news reports, movie reviews, and advertisements. Genre is defined on the way a text was created, the way it was edited and published, the register of language it uses, and the kind of audience to whom it is addressed [16].

Previous work on genre classification recognized that this task differs from topic-based categorization [16, 7]. A single genre, such as "written newswire" may encompass a range of topics, e.g., *sports, politics, crime, technology, economy and international events*. On the other hand, many articles on the same topics can be written in different genres. Therefore, the genre-topic mapping is many to many. Genre collections, such as ours discussed later, contain different genre covering with the same topic, *newswire, radio news and TV news*, in order to evaluate automated genre classification independent of topic classification. One way in which these two classification problems differ is that in general genre classification seldom exhibits individual features that highly predict a category, unlike topic classification, where words such as "umpire" and "RBI" directly predict the "baseball" category and indirectly the "sports" category.

Given the task of genre classification, the next questions are: How can we build accurate methods according to the characteristics of the genre data? Can we partially reuse the extensive body of work on topic classification? This paper explores aspects of these questions.

There have been many attempts to extract meaningful linguistic features to improve the prediction accuracy, such as POS tagging, parsing, number of punctuation, and layout features. However, many of those features (such as POS tagging or parsing) require high computational costs with little performance improvement; furthermore, for some text sources such as video, capitalization, punctuation and other such information are lost in the automatically speech-recognized transcript from the audio stream. Therefore it is useful to address genre classification using "bag of words" features only, which is the same for topic-based classification. Thus, instead of extracting other potential features, we focus on identifying the characteristics of the data in genre classification and propose suitable learning models accordingly.

Typically, most data for genre classification are collected from the web, through newsgroups, bulletin boards, and broadcast or printed news. They are multi-source, and consequently have different formats, different preferred vocabularies and often significantly different writing styles even for documents within one genre. Namely, the data are *heterogenous*. To illustrate this point, we provide an excerpt of two documents from the same genre, "bulletin-board", in our collected corpus:

– Example-1: *GSA announces weekly Happy Hours! Where: Skibo Coffeehouse When: Friday's 5-7pm What: Beer, Soda and Pizza Why: A chance to meet graduate students from all across campus. See you this Friday!*
– Example-2: *Hi guys, I don't know whether there is an informal party or not although different people kept saying there might be one... So if there is nothing, we can go to Cozumel tonight cuz there will be a live Latin band tonight starting at 9:30pm. But if there is anything else, then let me know.*

Heterogeneity is an important property shared by many other problems, such as scene classification and handwritten digit recognition. However, typical studies in topic-based classification assume homogenous data and tight distributions[1]. Extending classification for high-variance heterogeneous data is an interesting topic that has not been investigated, and is the primary focus of this paper.

Since the data are acquired from different sources and thus rather heterogeneous, a single classification model might not be able to explain all the training data accurately. One apparent solution to this problem is to divide the heterogeneous data into a set of relatively homogeneous partitions, train a classification model over each partition and combine the predictions of individual models. In

[1] The primary exception is Topic-Detection and Tracking (TDT) where multiple news sources are tracked in an on-line categorization task [1, 25].

this way, each sub-model captures only one aspect of the decision boundary. The idea of creating multiple models on the training data and combining the predictions of each model is essentially the ensemble approach, and there have been many studies on this subject. Several ensemble approaches have been successfully applied to text classification tasks, including boosting [8], Error-Correcting Output codes (ECOC) [6], hierarchical mixture model [22] and automated survey coding [12]. Alternative approaches such as stacking [23] and earlier meta-classifier approaches [2] do not partition the data, but rather combine classifiers each of which attempts to classify all data over the entire category space.

In this paper, we examine different ensemble methods for text classification. In particular, we propose an "input dependent latent variable" approach for model combination, which automatically directs each test example to the most appropriate classification model within the ensemble. We use this method as the framework to solve genre classification problems. Although our discussion is focused on multi-class classification framework, it is not difficult to extend to multi-label classification problems. The rest of the paper is organized as follows: in Section 2 we give an in-depth discussion of the popular ensemble approaches for topic-based text classification. Then we present our pairwise ensemble approach in Section 3. We compare our method with other ensemble methods on four datasets, including one artificial dataset, two genre datasets and one topic-based classification data. Finally, we give conclusion and hint at future work.

2 Popular Ensemble Approaches for Text Classification

Generally speaking, an ensemble approach involves two stages, namely model generation and model combination. In this section, we examine the model generation and model combination strategies in the popular ensemble approaches for the topic-based classification. Since genre classification also uses "bag of words" features, hopefully we can reuse some of the successful learning methods from topic classification to help genre classification.

Bagging involves a "bootstrap" procedure for model generation: each model is generated over a subset of the training examples using random sample with replacement (the sample size is equal to the size of the original training set). From a statistical point of view, this procedure asymptotically approximates the models sampled from the Bayesian posterior distribution. The model combination strategy for bagging is majority vote. Simple as it is, this strategy can reduce variance when combined with model generation strategies. Previous studies on bagging have shown that it is effective in reducing classification errors [4].

Boosting As a general approach to improving the effectiveness of learning, boosting [8] has been the subject of both theoretical analysis and practical applications. Unlike bagging, in which each model is generated independently, boosting forces the base classifier to focus on the misclassified examples in previous iterations. In this way, each new model can compensate for the weakness of previous models and thus correct the inductive bias gradually [17]. Applying boosting to text categorization tasks, Schapire and Singer evaluated AdaBoost on the benchmark corpus of Reuters news stories and obtained results comparable to Support Vector Machines and k-Nearest Neighbor methods [21], which are among the top classifiers for text classification evaluation [24, 14]. Empirical studies on boosting and bagging show that while both approaches can substantially improve accuracy, boosting exhibits greater benefits [19, 9]. Therefore, we provide only the results of boosting in our comparative experiments.

ECOC is an ensemble approach for solving multiclass categorization problems originally introduced by Dietterich and Bakiri[6]. It reduces a k-class classification problem into L ($L \leq k$) binary classification problems and combines the predictions of those L classifiers using the nearest codeword (for example, by Hamming distance). The *code matrix* R (an $k \times L$ matrix) defines how each sub-model is generated. There have been many code matrixes proposed, such as Dense matrix and BCH codes [20]. Recent work has demonstrated that ECOC offers improvement over the standard *one-against-all* method in text classification and provided theoretical evidence for the use of random codes [3, 11].

3 Pairwise Ensemble Approaches

From the discussion in section 2, we can see that most of those methods have complex model generation procedures and demonstrate considerable empirical improvement. However, they may not be the best choices for classification problems with heterogeneous data for two reasons: 1) In order to capture the heterogeneous characteristics of the data, it would be desirable to divide the training data into several relatively homogenous subsets. However, most algorithms do not intentionally do so. 2) The combination strategies are rather simple. To better solve the heterogenous classification problems, we propose the pairwise ensemble approach. The key idea of our algorithm is:

– build pairwise classifiers to intentionally divide the training data into relatively less heterogeneous sets so that each base classifier focuses on only one aspect of the decision boundary;
– combine the results using the "input-dependent latent variable" approach, which can consider the particular properties of each testing example and dynamically determine the appropriateness of each base classifiers.

3.1 Model Generation by Pairwise Classification

Since our data are quite heterogenous, it presents difficulties to the classical *one-against-all* method, which is implied in our experiment results in section 4. A natural idea would be applying pairwise classification method to discover the exact difference between each pair of genres and then combine the predictions of the individual classifiers. One big advantage of this approach is that each sub-classifier only need capture one local aspect of the training data while in the single model approach it has to fit all the aspects of the entire training data, which can average out important local distinctions.

Building pairwise classifier for multi-class classification problems is not a new idea and many attempts have been made to build ensemble approaches, such as ECOC [6], pairwise coupling [13], and round robin ensemble [10]. However, there has been little prior work on automatically combining individual pairwise classifier results in a meaningful way.

3.2 A General "Latent Variable" Approach for Combining Models

After the pairwise classifiers have been built, the remaining problem is how to combine the results. Linear combination methods, such as weighted voting, are inappropriate for the pairwise classification because each individual classifier only captures local information. One sub-classifier may be good for some examples, but not necessarily for all the testing data. Thus, a better strategy is to build a set of "gates" on top of the individual models and ask the "gate" to tell whether the corresponding model is good at capturing the particular patterns of the input test data. We would call this "input-dependent latent variable" because those gates can be thought as latent variables that determine the right models for each input data. Next, we give a formal description of this strategy.

Given the input data $\mathbf{x}$ and a set of ensemble models $M = \{m_1, m_2, \ldots, m_n\}$, our goal is to compute the posterior probability $P(y|\mathbf{x}, M)$. As shown in Figure-1, each gate, i.e. hidden variable, is responsible for choosing whether its corresponding classifier should be used to classify the input pattern. More precisely, let h_i stand for the hidden variable corresponding to the i_{th} classification model; the value of h_i can be 1 or 0, with 1 representing that the i_{th} model is selected for classifying the input example and 0 otherwise. By using the hidden variables, we can expand the posterior probability as a sum as follows:

$$P(y|\mathbf{x}, M) = \sum_{k_i \in \{0,1\}} P(y, h_1 = k_1, h_2 = k_2, \ldots, h_n = k_n | \mathbf{x}, M).$$

By assuming that the selection of a classification model is independent from the selection of another, we can simplify the joint probability as follows:

$$\begin{aligned} &P(y, h_1 = k_1, h_2 = k_2, \ldots, h_n = k_n | \mathbf{x}, M) \\ &= \prod_{i=1}^{n} P(h_i = k_i | \mathbf{x}, M) \times P(y | h_1 = k_1, \ldots, h_n = k_n, \mathbf{x}, M). \end{aligned}$$

Consider building an exponential model with a set of features $\{\ln P(y|\mathbf{x}, m_1), \ldots, \ln P(y|\mathbf{x}, m_n)\}$, then $P(y|h_1 = k_1, \ldots, h_n = k_n, \mathbf{x}, M) = \frac{1}{Z} \exp \sum_i \alpha_i \ln P(y|\mathbf{x}, m_i)$. To incorporate h_i into the equation above, we set α_i to k_i, with the intuition that the prediction of a model is given high weight if the model is suitable for the input pattern and low weight otherwise. In this way, we get

$$P(y|h_1 = k_1, \ldots, h_n = k_n, \mathbf{x}, M) \approx \frac{1}{Z} \prod_{i=1}^{n} P^{k_i}(y|\mathbf{x}, m_i). \tag{1}$$

We drop the normalization factor Z, rewriting the previous equation as a proportionality, and then the joint probability can be derived as follows:

$$P(y, h_1 = k_1, h_2 = k_2, \ldots, h_n = k_n | \mathbf{x}, M) \propto \prod_{i=1}^{n} P(h_i = k_i | \mathbf{x}, M) P^{k_i}(y|\mathbf{x}, m_i).$$

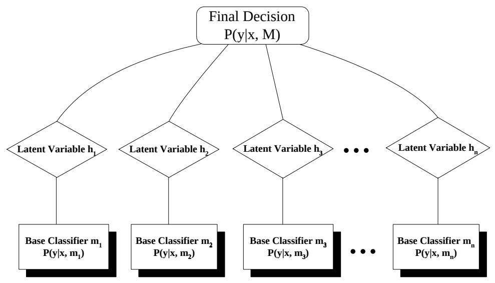

Fig. 1. The Structure of the Latent Variable Approach

Therefore

$$\begin{aligned} P(y|\mathbf{x}, M) &\propto \sum_{k_i \in \{0,1\}} \prod_{i=1}^{n} P(h_i = k_i|\mathbf{x}, M) P^{k_i}(y|\mathbf{x}, m_i) \\ &= \prod_{i=1}^{n} \sum_{k_i \in \{0,1\}} P(h_i = k_i|\mathbf{x}, M) P^{k_i}(y|\mathbf{x}, m_i) \\ &\propto \prod_{i=1}^{n} \{P(h_i = 1|\mathbf{x}, M) P(y|\mathbf{x}, m_i) + P(h_i = 0|\mathbf{x}, M)\}. \end{aligned}$$

By assuming $P(h_i = 0|\mathbf{x}, M) \to 1$ [2], we have

$$P(y|\mathbf{x}, M) \propto \prod_{i=1}^{n} \{P(h_i = 1|\mathbf{x}, M) P(y|\mathbf{x}, m_i) + 1\}.$$

In this way we can further simplify by expanding only to the first order and ignoring the high order terms that usually express the interaction between different models, which are usually very small in value . At last, we get the approximation:

$$P(y|\mathbf{x}, M) \propto \sum_{i=1}^{n} P(h_i = 1|\mathbf{x}, M) P(y|\mathbf{x}, m_i) \tag{2}$$

As indicated in (2), there are two major components: $P(h_i = 1|\mathbf{x}, M)$, i.e. the component describing how likely it is the i_{th} classifier should be used for classifying the input example, and $P(y|\mathbf{x}, m_i)$, i.e. the component determining the likelihood that class y is the true class label given the input $\mathbf{x}$ and the classification model m_i. At first glance, (2) looks very similar to the linear combination strategies except that the combination factor is $P(h_i = 1|\mathbf{x}, M)$. However, unlike the linear combination strategies whose combination weights is the same for all inputs, the weights in the latent variable approach are strongly connected with the input example by the conditional probability $P(h_i = 1|\mathbf{x}, M)$.

[2] During the developing process, we have made two assumptions ((1) and this one). In section 4 we will show that our approach demonstrates significant improvement over other methods even with those simplifying assumptions.

3.3 "Latent Variable" Approach for Pairwise Classifier

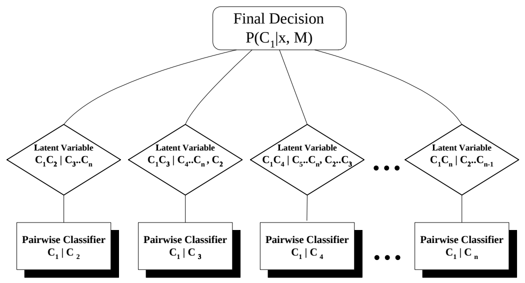
Final Decision
$P(C_1|x, M)$
Latent Variable $C_1C_2 | C_3..C_n$
Latent Variable $C_1C_3 | C_4..C_n, C_2$
Latent Variable $C_1C_4 | C_5..C_n, C_2..C_3$
Latent Variable $C_1C_n | C_2..C_{n-1}$
Pairwise Classifier $C_1 | C_2$
Pairwise Classifier $C_1 | C_3$
Pairwise Classifier $C_1 | C_4$
Pairwise Classifier $C_1 | C_n$

Fig. 2. The Structure of the Pairwise Ensemble Approach for Class C_1

Given the latent variable approach as the framework, the remaining problem is to estimate the conditional probability $P(h_i = 1|\mathbf{x}, M)$, i.e. the likelihood that the i_{th} model should be used for class prediction given the input $\mathbf{x}$. Since each individual classifier is a pairwise classifier to differentiate two classes, say C_i and C_j, a simple method to estimate $P(h_i = 1|\mathbf{x}, M)$ is to build a binary classifier on top of each base classifier to differentiate examples that belong to these two classes (C_i and C_j) and those that do not. The underlying idea is that the likelihood for a model to be used for classifying an input datum $\mathbf{x}$ is equal to the likelihood that $\mathbf{x}$ is similar with the examples to train the model.

To make it more clear, let n be the number of classes and $M_{C_i|C_j}$ represent the pairwise classifier to differentiate class C_i and C_j. On the top level, we will have another classifier $M_{C_i,C_j|\overline{C_{i,j}}}$ to differentiate whether the examples belong to one of classes C_i, C_j or not. Figure 2 shows the structure of the model for class C_1. Compared with (2), for the pairwise ensemble approach

$$P(y = C_1|\mathbf{x}, m_{1i}) = P(y = C_1|\mathbf{x}, y \in \{C_1, C_i\}, m_{1i}),$$
$$P(h_i = 1|\mathbf{x}, M_1) = P(y \in \{C_1, C_i\}|\mathbf{x}, M_1).$$

For each class C_i, we build a structure like this and compute the corresponding score of the test examples. For multiclass problem, typical in genre classification, we can assign the test example to either the class label with the highest score (R-cut) or the classes whose scores are above some preset thresholds (S-cut) [26]. In this way, our approach can be extended to multi-label classification problems by assigning all the class labels above a given threshold to each test instance.

3.4 Related Work and Discussion

Our work is related to several approaches, including hierarchical mixture of experts (HME) [15] and pairwise coupling [13]. HME uses similar ideas to dynamically determine the most appropriate model for testing examples. However, it

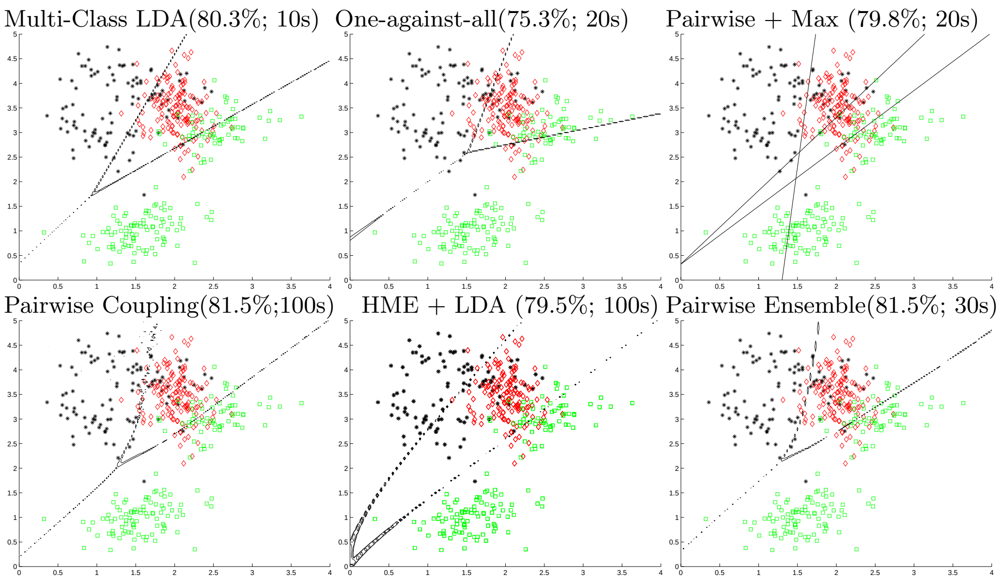

Fig. 3. Comparison of time and accuracy of Different Ensemble Approaches
Sub-title lists method, accuracy and estimated running time
$C1 \rightarrow *(black), C2 \rightarrow \sqcap(blue), C3 \rightarrow \diamond(red)$

requires much higher computational costs because it applies an EM algorithm to estimate the latent variables. Pairwise coupling also incurs high costs in the test phase due to its iterative search procedures. Therefore it would be difficult to directly apply those two methods to text classification problems. In order to provide a rough idea on the efficiency and effectiveness of those ensemble methods, we followed the experiments in [13] and generated a synthetic dataset of 3 classes with the data in each class generated from a mixture of Guassians. We use Linear Discriminative Analysis (LDA) as base classifiers. The results and the decision boundary are shown in Figure 3. From the results, we can see that pairwise ensemble and pairwise coupling are the best in terms of accuracy, and our method is much faster.

4 Empirical Validation

In our experiments we chose two datasets for genre-based classification evaluation. Collection I consists of 12,259 documents from 10 genres (for details see Table 1). We split the corpus into a training set of 9,236 documents and a test set of 3,023 documents. The Radio-news, TV-news and Newswire are part of the TDT2 [1] and we extracted documents from the same time period in order to ensure similar contents and thus minimize the information provided due to different topics instead of different genres. The rest of the documents were collected from the web. Collection II, provided by Nigel Dewdney, consists of about 3,950 documents from 17 genres (see Table 2 for details). We split the corpus into a training set of 3,000 documents and a test set of 950 documents. Compared with collection I, collection II contains more genres, but they are easier to distinguish.

We pre-processed the documents, including down-casing, tokenization, removal of punctuation and stop words, stemming and supervised statistical features selection using the $\chi^2_$max criterion. The optimal feature set size was chosen separately by 10-fold cross validation. Finally we chose 14,000 features and 10,000 features for Collection I and Collection II, respectively. Document vectors based on these feature sets were computed using the SMART *ltc* version of TF-IDF term weighting [5]. For the evaluation metric, we used error rate and a common effectiveness measure, F_1, defined to be [26]: $F_1 = \frac{2rp}{r+p}$ where F_1 is the *harmonic average* of precision p and recall r. To measure overall effectiveness we use both the *micro-average* (effectiveness computed from the sum of per-category contingency tables) and the *macro-average* (unweighted average of effectiveness across all categories).

Genre	Newswire	Radio	TV	Message	Ad.	FAQ	Politics	Bulletin	Review	Search Result
Number	2082	1810	1145	1106	1091	1063	999	998	996	969

Table 1. Document Distribution in Collection I

Genre	Jokes	Recipe	Quotes	Tips	Newspages	Advice	Poetry	Horoscopes	Conference
Number	315	302	293	270	252	243	231	223	211

Genre	Resume	Company	Personal	Interview	Article	Search	Homepages	Classified
Number	203	202	200	201	201	201	202	200

Table 2. Document Distribution in Collection II

4.1 Experimental Results

In our experiments we used Support Vector Machines, one of the most powerful classifiers in previous text classification evaluation [26], relying on the SVM_{Light} package [14]. Table 3 & 4 shows the results by different ensemble approaches and their improvement over the baseline on Collection I & II respectively.

We use the result of SVM with linear kernel, without any ensemble methods as baseline. For boosting, we use the AdaBoost algorithm with SVM, tuned for the optimal number of training iterations and report the best results (the corresponding training iteration is 10 and 5 respectively for collection I and II). For ECOC, we use SVM as the base classifier and apply a 63-bit random coding for both collections, which is also used in [11] for their experiments. SVM is used for both base classifiers and the top level gate classifier in pairwise ensemble.

From the results, we can see the pairwise ensemble approach performs consistently the best among the four methods in terms of error minimization and both Micro-F_1 and Macro-F_1 measurement. Boosting SVM decreased performance over the baseline for both collections, which is a sign of overfitting. In fact, we tried boosting other classifiers, such as Decision Tree and Naive Bayes. Although boosting SVM deteriorates the performance, it gives the best result compared to

Method	**Micro-Avg F_1**		**Macro-Avg F_1**		**Error Rate**
One-against-all + SVM	0.8757	n/a	0.8780	n/a	11.6%
Pairwise Ensemble + SVM	**0.8965**	+2.3%	**0.9003**	+2.5%	10.4%
Boosting + SVM	0.8695	-0.7%	0.8726	-0.6%	12.4%
ECOC + SVM	0.8720	-0.4%	0.8758	-0.3%	12.8%

Table 3. Comparison of Results for Collection I

Method	**Micro-Avg F_1**		**Macro-Avg F_1**		**Error Rate**
One-against-all + SVM	0.9013	n/a	0.8755	n/a	9.1%
Pairwise Ensemble + SVM	**0.9495**	+5.3%	**0.9432**	+7.7%	5.1%
Boosting + SVM	0.8903	-1.2%	0.8620	-1.5%	9.9%
ECOC + SVM	0.9126	+1.3%	0.9026	+3.1%	8.7%

Table 4. Comparison of Results for Collection II

other boosted classifiers. ECOC method achieved some improvement on Collection II but decreased the performance for the other collection. This implies that ECOC is not a generally effective method to improve classification accuracy.

In order to more clearly evaluate automated genre classification, independent of topic classification, we listed in Table 5 the detailed results of three genres in Collection I, i.e., *Newswire, Radio-news and TV-news*, which have been intentionally collected on the same topics to minimize the information provided due to different topics instead of different genres. From the results, we can see that the performance on those three categories is much lower than the average result over the whole collection with ten categories. This implies that it is more difficult and challenging to distinguish genre within the same topic. On the other hand, our approach achieves the best performance on all the three categories, especially for TV-news, which improves abut 9% over the baseline in F_1 measure.

	Newswire	Radio-news	TV-news	Macro-Avg F_1
One-against-all + SVM	0.9297	0.7635	0.7572	0.8168
Pairwise Ensemble + SVM	0.9337	0.7838	0.8240	0.8472
Boosting+SVM	0.9327	0.7529	0.7603	0.8153
ECOC+SVM	0.8894	0.7669	0.8073	0.8212

Table 5. Comparison Results of Three Genres within the Same Topic (F_1 measure)

4.2 Extension for Topic-based Text Classification

We have shown that the pairwise ensemble approach is effective to improve the performance of genre classification. Since we use only word features for genre classification, which is the same with the topic-based classification, it is a natural question to ask whether our method is also good for topic-based text classification. To answer the question, we tested our methods on the Newsgroups dataset [18], one of commonly used datasets for text classification. The dataset contains 19,997 documents evenly distributed across 20 classes. We used the cleaned-up

Method	Micro-Avg F_1		Macro-Avg F_1		Error Rate
One-against-all + SVM	0.9009	n/a	0.8941	n/a	9.3%
Pairwise Ensemble + SVM	**0.9333**	+3.6%	**0.9257**	+3.5%	6.8%
Boosting + SVM	0.9020	+0.1%	0.8967	+0.3%	9.1%
ECOC + SVM	0.9159	+1.7%	0.9025	+1.0%	8.4%

Table 6. Comparison of Results for 20Newsgroup

version of the dataset[3], removed stop words as well as the words that occur only once, with the final vocabulary size being about 60,000. We randomly select 80% of the documents per class for training and the remaining 20% for testing (15199 training documents and 3628 test documents). This is the same pre-processing and splitting as in the McCallum and Nigam experiments [18].

Table 6 lists the results of comparing different ensemble approaches on the newsgroup dataset. For boosting, the training iteration was 10 by cross-validation and all other parameter settings are the same with previous experiments. The results imply that the pairwise ensemble approach works well for this text classification dataset, in fact significantly better that baseline SVM or boosting SVM.

5 Conclusion and Future Work

In this paper, we identified the heterogeneity of genre data and presented our new pairwise ensemble approach to capture this characteristic. Empirical studies on two genre datasets and one topic-based datasets show that our method achieved the best performance among all the popular ensemble approaches we have tried, including boosting and ECOC. However, is pairwise ensemble truly dominant in general? Answering that question would require much larger scale empirical studies, but is definitely an important issue. Another line of research involves discovering the limitations of pairwise ensemble, such as the computational tractability as the category space grows and potential paucity of data to train all pairwise classifiers. One solution would be selecting only category pairs with sufficient training data and smoothing the decisions via the baseline classifier. Empirical validation for these extensions would be a natural next step.

6 Acknowledgments

This work presented in this paper was funded in part by the National Science Foundation Award # EIA-0225656. Any opinions or findings in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

1. J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. 1998.

[3] This cleaned-up version is downloaded from http://www.ai.mit.edu/people/jrennie/ecoc_svm

2. P. N. Bennett, S. T. Dumais, and E. Horvitz. Probabilistic combination of text classifiers using reliability indicators: Models and results. In *SIGIR'02*, 2002.
3. A. Berger. Error-correcting output coding for text classification. In *IJCAI'99: Workshop on machine learning for information filtering*, 1999.
4. L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
5. C. Buckley, G. Salton, and J. Allan. The effect of adding relevance information in a relevance feedback environment. In *SIGIR-94*, pages 292–300, 1994.
6. T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.
7. A. Finn, N. Kushmerick, and B. Smyth. Genre classification and domain transfer for information filtering. In *Proceedings of ECIR-02*, 2002.
8. Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.
9. Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.
10. J. Fürnkranz. Round robin rule learning. In *Proceedings of the 18th International Conference on Machine Learning (ICML-01)*, pages 146–153, 2001.
11. R. Ghani. Using error-correcting codes for text classification. In *Proceedings of 17th International Conference on Machine Learning*, pages 303–310, 2000.
12. D. Giorgetti and F. Sebastiani. Multiclass text categorization for automated survey coding. In *ACM Symposium on Applied Computing*, pages 798–802, 2003.
13. T. Hastie and R. Tibshirani. Classification by pairwise coupling. In *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.
14. T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *European Conference on Machine Learning (ECML)*, pages 137–142, Berlin, 1998. Springer.
15. M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.
16. B. Kessler, G. Nunberg, and H. Schütze. Automatic detection of text genre. In *Proceedings of the Thirty-Fifth ACL and EACL*, pages 32–38, 1997.
17. Y. Liu, Y. Yang, and J. Carbonell. Boosting to correct the inductive bias for text classification. In *Proc. of CIKM'02*, 2002.
18. A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.
19. J. R. Quinlan. Bagging, boosting, and c4.5. *Proceedings of the 13th National Conference on Artifitial Intelligence on Machine Learning*, pages 322–330, 1996.
20. J. Rennie. Improving multi-class text classification with support vector machine. *Master's thesis, Massachusetts Institute of Technology*, 2001.
21. R. Schapire and Y. Singer. Boosttexter: Aboosting-based system for text categorization. In *Machine Learning*, volume 39(1/3), pages 135–168, 2000.
22. K. Toutanova, F. Chen, K. Popat, and T. Hofmann. Text classification in a hierarchical mixture model for small training sets. In *Proc. of CIKM'01*, 2001.
23. D. Wolpert. Stacked generalization. *Neural Networks*, pages 241–259, 1992.
24. Y. Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1(1/2):67–88, 1999.
25. Y. Yang, J. Carbonell, R. Brown, J. Lafferty, T. Pierce, and T. Ault. Multi-strategy learning for topic detection and tracking. In *TDT99 book*. Kluwer Academic Press.
26. Y. Yang and X. Liu. A re-examination of text categorization methods. In *SIGIR'99*, pages 42–49, 1999.