Text Mining Using An Ensemble Classifier

B.Meena Preethi[1], R.Sruthi[2]
[1]Associate Professor, Dept Of Software Systems
[2] Dept Of Software Systems
[1, 2] Sri Krishna Arts And Science College, Coimbatore

Abstract- *Text mining is the progression of determining and analyzing hefty amounts of formless text data assisted by software that can identify concepts, patterns, topics, keywords and other attributes in the data. Expert, labeled data is necessary for effectively applying machine learning techniques to real-world text classification problems. High rate and attempt involved in labeling the data, the quantity of labeled data is petite compared to the amount of unlabeled data. An ensemble method is a machine learning technique that combines several base models in order to produce one optimal predictive model. Ensemble methods, with respect to categorization algorithms are moderately new techniques. Ensemble learning algorithms are broad methods that raise the exactness of analytical or classification models such as decision trees, artificial neural networks. Text mining using an ensemble classifier is used to boost the accuracy of data.*

Keywords- Text Mining, ensemble methods, co-training algorithm

I. INTRODUCTION

Organized methods for inducing exact text classifiers rely on the accessibility of large amounts of labeled data. Though, in numerous applications, because of the high price and attempt concerned in labeling the data, the total of labeled data is small compared to the amount of unlabeled data. Because of this growing gap between the rate of acquirement of textual data and the rate of guide labeling, there is a significant interest in semi supervised algorithms that can exploit large amounts of unlabeled data together with limited amounts of labeled data in training text classifiers [1].Ensemble methods increase overall model accuracy while cross validation techniques increase the precision of model error estimation [2]. The improved accurateness of an ensemble, because of model variance diminution and to a smaller extent bias reduction, is based on the plain but powerful process of group averaging or majority vote. The diversity necessity brings diverse basis of information to the decision procedure, which expands the resolution space of achievable solutions. A cluster decision cannot be extra precise if all group members' desire or recommend the similar solution. The independence of action obligation mitigates the prospect of a herd approach where group members sway or authority other members towards one definite solution.

This wealth of data has the potential to considerably adjust the technique that we access, process ,and utilize information. Progressively more, the confront has become one of trying to construct logic of the information available and organize it in such a way that it can used to maximum benefit. A great deal of the complexity is that so much of the functional data being generated by users online is generated in a announcement medium that is easiest for humans to create and process, namely natural language. Because of this, much of the confront lies in increasing computer software that can process written natural language, aggregate it, organize it, and present it back to humans in meaningful and, often, more succinct ways. One of the requirements for co-training to work is that the data can be represented using two different "views" of features on which two separate classifiers are trained. Unlabeled information are then labeled with these two classifiers. Service improvement framework, which includes GLOW and SMOKE words, that integrates traditional text mining methods with innovative ensemble learning techniques. The accuracy of data is achieved using ensemble classifier.

II. TEXT MINING

Text Mining is also known as Text Data Mining. The purpose is too unstructured information, extract meaningful numeric indices from the text. Thus, make the information contained in the text accessible to the various algorithms. Information can extract to derive summaries contained in the documents.

Text Analytics, also known as text mining, is the process of examining large collections of written resources to generate new information, and to transform the unstructured text into structured data for use in further analysis. Text mining identifies facts, relationships and assertions that would otherwise remain buried in the mass of textual big data.

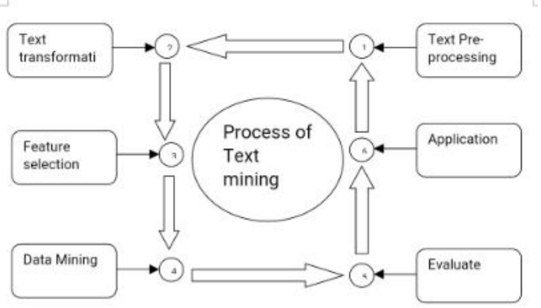


FIG1: TEXT MINING PROCESS

III. TYPES OF ENSEMBLE METHODS

1. BAGGING

Baggingor bootstrap aggregating. Bagging gets its name because it combines Bootstrapping and Aggregation to form one ensemble model. Given a sample of data, multiple bootstrapped subsamples are pulled. A Decision Tree is formed on each of the bootstrapped subsamples. After each subsample Decision Tree has been formed, an algorithm is used to aggregate over the Decision Trees to form the most efficient predictor [3]. The image below will help explain:

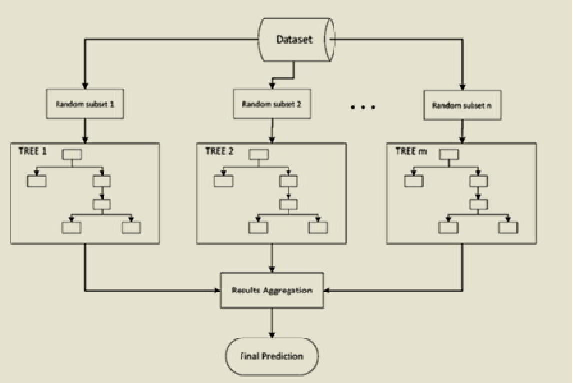
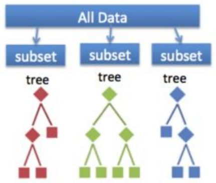


FIG 2: BAGGING

2. RANDOM FOREST MODELS

Random Forest Models can be thought of as **BAGG**ing, with a slight tweak. When deciding where to split and how to make decisions, BAGGed Decision Trees have the full disposal of features to choose from. Therefore, although the bootstrapped samples may be slightly different, the data is largely going to break off at the same features throughout each model. In contrary, Random Forest models decide where to split based on a random selection of features. Rather than splitting at similar features at each node throughout, Random Forest models implement a level of differentiation because each tree will split based on different features. This level of differentiation provides a greater ensemble to aggregate over, ergo producing a more accurate predictor. Refer to the image for a better understanding.



A random forest takes a random subset of features from the data, and creates n random trees from each subset. Trees are aggregated together at end.

FIG 3: RANDOM FOREST

3. BOOSTING

Boosting involves incrementally building an ensemble by training each new model instance to emphasize the training instances that previous models mis-classified. In some cases, boosting has been shown to yield better accuracy than bagging, but it also tends to be more likely to over-fit the training data.

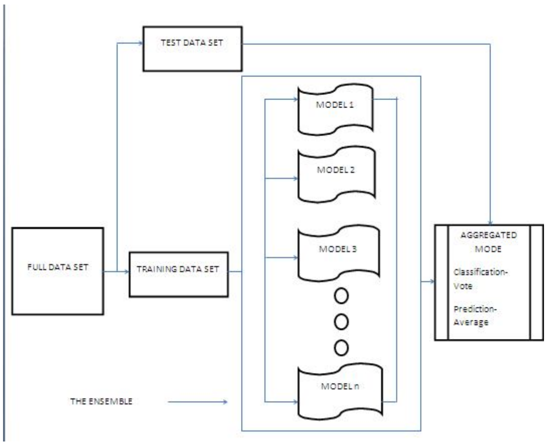


FIG 4: ENSEMBLE METHOD

ii) ENSEMBLE CLASSIFICATION USING MEAN CO-ASSOCIATION MATRIX

The MECAC (Ensemble Classification using Mean Co-Association Matrix) algorithm uses **the mean co-association matrix**, usually used in consensual clustering problems [4]. MECAC – to build an ensemble of classifiers that has two advantages to other ensemble methods: 1) it can be run using parallel computing, saving processing time and 2) it can extract important statistics from the obtained clusters. It uses the mean co-association matrix to solve binary TC problems. Our experiments revealed that our framework performed, on average, 2.04% better than the best individual classifier on the tested datasets. These results were statistically validated for a significance level of 0.05 using the Friedman Test.

Four state-of-art classifiers for single-label TC to carry in our experiments: Support Vector Machines with a linear kernel (SVM-linear), k Nearest Neighbors (kNN), Naïve Bayes (NB) and Neural Networks (NNET). Firstly, we build a baseline ensemble method (ENS-b) for comparison against the ensemble approach we propose. ENS-b uses the majority class considered among the base classifiers (we used four models, one model per algorithm). Secondly, we used the same base learners to build two ensembles using MECAC: ENS1, that used all the four models, and ENS2 that used all except NB. Finally, we compared the four base learners, ENS-b, ENS1 and ENS2 using three performance metrics: macro avg./micro avg. F1-measure and Cohen- Kappa to classify document collections of Reuters-21578 dataset [5].

Step 1 – The Classifiers Training

A set of classifiers is generated by applying k classification algorithms to our training set *X*. π = [π1, π2 … π3] contains the class determined by the classifiers to our test set."

Step 2 – The calculus of the Agreement Matrix

We present an algorithm that creates **a new distance measurement based on the agreement between the k classifiers**. Let *M* (*s* x *s*) be a quadratic matrix as large as the number of test documents. The values in the matrix measure the agreement between the classifiers to categorize equally both documents. The **mean coassociation matrix** [8], represents the classification agreement between all classifiers.
The values in the matrix are obtained as follows:

$$M\ (i, j) = \{\ 2^{a-1} \text{ if } a > 0 \\ 0 \text{ if } a = 0 \\ , i, j\ €\{1,\ldots.,s\}$$

where *a* is the number of classifiers that classified equally the documents *i* and *j*, independently on the true class of both documents. This matrix measures the agreement of the classifiers to label equally each pair of documents. This information is directly about the similarity between each pair of documents - then the category is calculated based on it. **The weights** (the pow used to calculate the agreement instead of a simple sum) **were introduced to enhance the agreement achieved between all classifiers**: it is measured exponentially to express its consensus relevance. **This weighted measure is one of the main contributions of this work because it innovates the calculus of the distance between text documents for binary TC** (the simple sum proposed in [8] performs worst in the current context). Such distance highlights the agreement between the classifiers (i.e. the similarity between the documents). After its normalization (we divide all the values in the matrix for its maximum value), it is possible to transform the matrix *M* into the quadratic matrix D (*s* x *s*), as follows:

$$D = 1 – M/ma$$

where *ma* is the previously referred maximum.

Step 3 – The Document Clustering

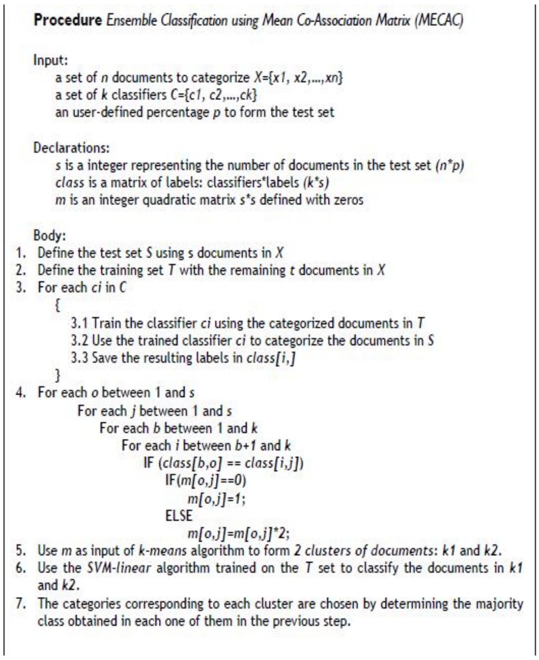
We use the matrix D as input for a clustering algorithm of interest like *k-means.* Wesplit the test set into 2 unlabeled partitions because this is a binary classification problem[6,7,8].

IV. THE CO-TRAINING ALGORITHM

A. The Co-Training Algorithm

The original co-training algorithm assumes that two independent views of the data are available and are used to train two classifiers. The classifiers are trained on a set of labeled training instances *L* and are used to classify a set of unlabeled training instances *U* . Then, iteratively, each classifier selects a few instances from *U* of which it is most certain (for each of the target classes) and adds them to *L*. For a classifier, the degree of certainty is determined by the probabilities assigned to each unlabeled instance for belonging to class c. The instances are ordered by their probabilities, and we select the top *X* ranking instances. We then add these instances to the labeled training instances set *L*. The label assigned to each instance is the one presumed by the classifier. The intuition behind this method is that the two classifiers are trained on two different views of the data, and therefore one classifier can improve the performance of the other. As one classifier feeds the other instances it may not be certain how to classify, the latter receives instances it has problems classifying on its own, and thus the overall performance improves. The basic (original) co-training algorithm that we use as our baseline is shown in Algorithm 1. The input to the algorithm is a labeled set of instances (*L*), an unlabeled set of instances (*U*), a test set (*T*), a classification algorithm (*I*), and the number of iterations of the co-training (*n*).

The product of the algorithm is a classification model whose training set consists of the extended training set (*ET*). This set consists of the initial labeled set (*L*) and a portion of the unlabeled set that was labeled throughout the training iterations and added to (*L*). Upon labeling, these newlylabeled instances are removed from the unlabeled set. In addition, the final classificationmodel produced by the co-training process completely ignores the knowledge represented by the various classifiers that were created “along the way.” To address these limitations, next we present our proposed co-training algorithm.[14]



```
Procedure Ensemble Classification using Mean Co-Association Matrix (MECAC)

Input:
    a set of n documents to categorize X={x1, x2,...,xn}
    a set of k classifiers C={c1, c2,...,ck}
    an user-defined percentage p to form the test set

Declarations:
    s is a integer representing the number of documents in the test set (n*p)
    class is a matrix of labels: classifiers*labels (k*s)
    m is an integer quadratic matrix s*s defined with zeros

Body:
1.  Define the test set S using s documents in X
2.  Define the training set T with the remaining t documents in X
3.  For each ci in C
       {
          3.1 Train the classifier ci using the categorized documents in T
          3.2 Use the trained classifier ci to categorize the documents in S
          3.3 Save the resulting labels in class[i,]
       }
4.  For each o between 1 and s
          For each j between 1 and s
             For each b between 1 and k
                For each i between b+1 and k
                   IF (class[b,o] == class[i,j])
                      IF(m[o,j]==0)
                         m[o,j]=1;
                      ELSE
                         m[o,j]=m[o,j]*2;
5.  Use m as input of k-means algorithm to form 2 clusters of documents: k1 and k2.
6.  Use the SVM-linear algorithm trained on the T set to classify the documents in k1
    and k2.
7.  The categories corresponding to each cluster are chosen by determining the majority
    class obtained in each one of them in the previous step.
```

FIG 5: Pseudo code of MECAC, the proposed ensemble methodology

B. **Ensemble of Co-Training Models**:

One of the prerequisites to the successful application of ensemble learning is diversity. The various classification models partaking in the ensemble need to offer varying “perspectives” (i.e., classifications or levels of confidence) regarding the labels of the instances of the training and validation sets [9]. The success of our proposed approach depends on the co-training models generated throughout the various iterations to possess this trait. To determine whether this is the case, we review two previous studies examining this subject. Mihalcea [10] analyzed the performance of the co-training algorithm throughout the training process in an attempt to determine the optimal number of training iterations. The analysis presented in this study shows significant fluctuations in precision for consecutive iterations, a fact that leads us to conclude that themodels are significantly different and therefore could be useful in an ensemble setting. Another study that supports this conclusion is that of Katz et al. [11], who demonstrate that different strategies for selecting additional instances have significant performance on the performance (i.e., the classification model) of the co-training algorithm. Intuitively, this conclusion is to be expected. Given the fact that the classification models generated during the co-training process are created using only a small number of instances, the addition of even a few instances is very significant percentage -wise. These changes are likely to be even more significant for classifiers such as decision trees, which utilize a hierarchical ordering of features with specific “split points” for each attribute (e.g., “if *x* < 4. take the left branch and otherwise the right”).

C. Co-training

Co-training, originally developed by Blum and Mitchell [12], is a semisupervised learning method designed to tackle these types of scenarios. Specifically, in addition to a small labeled training set, the co-training algorithm assumes that a large, unlabeled training set is available. One of therequirements for co-training towork is that the data can be represented using two different “views”of features on which two separate classifiers are trained. Unlabeled data are then labeled with these two classifiers. Iteratively, each classifier selects a few unlabeled samples for which it hasthe highest level of certainty in classification. These samples (a few from each class) are added to the labeled training set with the labels predicted by these classifiers in such a way that each classifier “trains” the other by providing it with samples that it may have difficulty classifying on its own. This process continues until some stopping criterion is met (e.g., until a predefined number of iterations is reached or all unlabeled data are used)[13].

V. CONCLUSION

It can be concluded from this project that if text mining using ensemble classification is used for large amounts of text documents, the results will be accurate and efficient. It will be very easy for the users to understand. Co-training algorithms, which make use of unlabeled data to improve classification, have proven to be very effective in such cases. Generally, co-training algorithms work by using two classifiers, trained on two different views of the data, to label large amounts of unlabeled data. Doing so can help minimize the human effort required for labeling new data, as well as improve classification performance. In this article, we propose an ensemblebased co-training approach that uses an ensemble of classifiers from different training iterations to improve labeling accuracy.Text mining using ensemble classifier is used to make the text accurate and efficient.

REFERENCE

[1] X. Zhu and A. B. Goldberg. 2009. *Introduction to Semi-Supervised Learning*. Morgan & Claypool.

[2] Kantardzic, M. (2011). Data Mining: Concepts, Models, Methods, and Algorithms. Hoboken, N.J., John Wiley: IEEE Press.

[3] https://towardsdatascience.com/ensemble-methods-in-machine-learning-what-are-they-and-why-use-them-68ec3f9fef5f

[4] Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data Machine Learning 52, 91–118 (2003)

[5] 5.http://www.daviddlewis.com/resources/testcollections/reuters21578/

[6] Yang, Y., Liu, X.: A Re-Examination of Text Categorization Methods. In: 22nd Annual International ACM SIGIR Conference on Research and Development in InformationRetrieval, pp. 42–49 (1999)

[7] Colas, F., Brazdil, P.: Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks. In: Artificial Intelligence in Theory and Practice, pp. 169–178(2006)

[8] Khan, A., Baharudin, B., Lee, L., Khan, K.: A Review of Machine Learning Algorithms for Text-Documents Classification. Journal of Advances in Information Technology 1(2010)

[9] C. M. Christoudias, R. Urtasun, and T. Darrell. 2008. Multi-view learning in the presence of view disagreement. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI'08)*. 88-96.

[10] Rada Mihalcea. 2004. Co-training and self-training for word sense disambiguation. In *Proceedings of the HLT-NAACL 2004 Workshop: 8th Conference on Computational Natural Language Learning (CoNLL'04)*. 33–40.

[11] Gilad Katz, Asaf Shabtai, and Lior Rokach. 2014. Adapted features and instance selection for improving co-training.In *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*. Springer, 81–100.

[12] A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT'98)*. ACM, New York, NY, 92–100. DOI:http://dx.doi. org/10.1145/279943.279962

[13] Vertical Ensemble Co-Training for Text Classification GILAD KATZ, Ben-Gurion University of the Negev,CORNELIA CARAGEA, University of North Texas ASAF SHABTAI, Ben-Gurion University of the Negev

[14] Text Categorization Using an Ensemble Classifier Based on a Mean Co-association Matrix Luís Moreira-Matias1,2, João Mendes-Moreira1,2, João Gama2,3, and Pavel Brazdil 2,3

[15] Sentiment Mining Using Ensemble Classification Models Conference Paper · January 2008