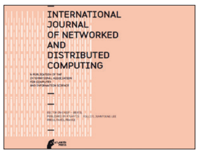
International Journal of Networked and Distributed Computing
Vol. **6**(**4**); September (2018), *pp.* 204–209
DOI: 10.2991/ijndc.2018.6.4.2; ISSN 2211-7946
https://www.atlantis-press.com/journals/ijndc



Social Media Comment Management using SMOTE and Random Forest Algorithms

Nuanwan Soonthornphisaj*, Taratep Sira-Aksorn, Pornchanok Suksankawanich

Department of Computer Science, Faculty of Science, Kasetsart University, Bangkok, Thailand

ARTICLE INFO

Article History
Received 27 June 2018
Accepted 20 September 2018

Keywords
Social media
imbalanced dataset
Random Forest
SMOTE

ABSTRACT

Comment posting is one of the main features found in social media. Comment responses enrich the social network function especially for how-to content. In this work, we focus on cooking video clips which are popular among users. Various questions found in comments need to be clustered in order to facilitate the clip owners to effectively provide responses for those viewers. We applied machine learning algorithms to learn and classified comments into predefined classes. Then the density-based clustering algorithm, for density-based spatial clustering of applications with noise, is applied to cluster the content of Comment. The experimental result show that using Random Forest with SMOTE provides the best performance. We got 95% of the average performance measured in term of F_1-measure. Furthermore, we implement the incremental learning system via an online application that can automatically retrieve and organize video clip's comment into categories.

1. INTRODUCTION

Food and recipes are one of the preferred content categories found in YouTube channels. Most of user's comments for how-to clips are in question forms related to Ingredients, Cooking tools and techniques. Even though the platform allows the communication among viewers but the comments has not yet been organized. The clip owner needs to interact with those comments however there are quite a number of comments asking the same questions whilst the owner must take his time to keep answering these comments. If we can manage the comment in such a way that the same category of question is grouped with the response. It will also benefit other viewers to explore frequently asked question based on category.

In this work, we collect posted comments from cooking video clips using YouTube API then we use ensemble learning method to learn from the training set and classify the comments into predefined class. The comment can be categorized into four classes which are: (1) How-to, (2) Ingredients, (3) Cooking tools and (4) Others. Since the dataset is imbalanced therefore, the imbalance class problem is solved by oversampling method.

The rest of paper is organized as follows. Section 2 gives the brief background of algorithms and related works. Section 3 provides the detail of our proposed method. Then all Experimental Results are shown in Section 4. Section 5 shows detail about application and the conclusion is given in the final section.

* *Corresponding author. Email: fscinws@ku.ac.th*

2. BACKGROUND

2.1. Random Forest

Our research deals with text classification problem where a dataset of comments must be categorized into predefined classes. Feature set in this domain is a word occurrence that lead to dimensionality problem. Therefore an ensemble decision tree learning method so called Random Forest is supposed to be an appropriated method since it produce diversity by training on different subsets of the data and feature sets. In machine learning point of view, diversity is the key to construct good ensembles.

Random Forests is introduced by Breiman [1], where he combines Bagging with random feature selection for decision trees. Each learner of the ensemble is trained on a bootstrap replicate as in Bagging. Decision trees are then grown by selecting the feature to split on at each node from *F* randomly selected features. After generating large number of trees, the final classification result is obtained from the majority vote. We call these procedures as Random Forests (see Table 1).

Dietterich [2] recommends Random Forests as the method of choice for decision trees, as it compares favorably to AdaBoost [3] and works well even with the presence of noise. His empirical research shows that AdaBoost is not a good choice in the noisy domain since it iteratively increases the weights on the instances most recently misclassified. Instances having incorrect class labels will persist in being misclassified. Then, AdaBoost will concentrate increasing weight on these noisy instances and become warped [1]. The Random Forest procedures do not concentrate weight on any

Table 1 Random Forest algorithm

```
Algorithm: Random Forest

Input: A training set $S: = (x_1, y_1), \ldots ,(x_n, y_n)$
  Features set, $F$
  Number of trees in forest, $B$
Output: Hypothesis
Begin
    $H \leftarrow \varnothing$
    for $i \in 1, \ldots , B$ do
      $S^{(i)} \leftarrow$ A bootstrap sample from $S$
      $h_i \leftarrow$ Randomized Tree Learn $(S^{(i)}, F)$
      $H \leftarrow H \cup \{h_i\}$
    endfor
  return $H$
end
Function: Randomized Tree Learn($S$, $F$)
  At each node:
    $f \leftarrow$ very small subset of $F$
    Split on best feature in $f$
  return The learned tree
end function
```

Table 2 Comment Classification algorithm

```
Algorithm: Comment Classification

Input
  dataset
Begin
  for each instance in dataset do
    comment = preprocessing()
    vector: = featureExtract(comment)
    add vector and label to vectors
  endfor
model: = Random Forest with SMOTE (SMOTE(trainset))
result: = model.predict(testset)
end
```

subset of the instances and the noise effect is smaller. Furthermore, the contribution of ensemble method is extensively explored in Melville and Mooney [4].

2.2. SMOTE

We found that our data set is imbalanced in terms of number of instances in each class see Table 5. The imbalanced data set affects the classification performance. The classification on imbalanced data causes problems because learning algorithm is sensitive to the number of training instances. The classification performance of majority class is higher than that of minority class. The problem is that the majority class is not the target class to be learned by the algorithm. In this research, class Others contains general comments such as admiring comments or comments that need no feedback from the clip owner. This class has higher number of comments than the target classes (How-to, Ingredients, Tools). Therefore we need the algorithm to manipulate the training set by increasing the size of the minority class either duplicates or interpolates minority instances. One of the famous oversampling method is the SMOTE algorithm [5]. SMOTE increases the number of minority class instances using interpolation method. The algorithm starts from searching for the k-nearest neighbors for each minority instance, then for each neighbor, it randomly selects a point from the line connecting the neighbor and the instance itself. This algorithm is able to generate synthetic instances rather than duplicate minority class instances; therefore, the overfitting problem can be avoided (see Table 3 for detail).

2.3. Density-Based Spatial Clustering of Applications with Noise

DBSCAN stands for density-based spatial clustering of applications with noise (Table 4). It finds core instance of high density and expands the cluster. The advantage of DBSCAN is that it does not require the number of clusters as a parameter. The algorithm was initially deployed to solve spatial data [6], however it was applied to solve many textual datasets such as Web page [7] and Newsgroup [8].

Table 3 SMOTE algorithm

```
Algorithm: SMOTE

Input: Number of minority class samples, $T$
  Amount of SMOTE, $N$%
  Number of nearest neighbors, $k$
Output: $(N/100) * T$ synthetic minority class samples
// If $N$ is less than 100%, randomize the minority class samples as only a
  random percent of them will be SMOTE
Begin
  if $N < 100$ then
    Randomize the $T$ minority class samples
    $T = (N/100) * T$
    $N = 100$
  endif
$N = (int)(N/100)$ // The amount of SMOTE is assumed to be in integer
  multiples of 100.
$k$ = Number of nearest neighbors
numattrs = Number of attributes
Sample[ ][ ]: Array for original minority class samples
newindex: Keeps a count of number of synthetic samples generated,
  initialized to 0
Synthetic[ ][ ]: Array for synthetic samples
// Compute $k$-nearest neighbors for each minority class sample only
  for $i \leftarrow 1$ to $T$
    Compute $k$-nearest neighbors for $i$, and save the indices in the nnarray
    Populate($N$, $i$, nnarray)
endfor
end

Populate(N, i, nnarray)
//Function to generate the synthetic samples//
  while $N \neq 0$
    nn = Random()
    for attr $\leftarrow$ 1 to numattrs
      dif = Sample[nnarray[nn]][attr] − Sample[i][attr]
      gap = random number between 0 and 1
      Synthetic[newindex][attr] = Sample[$i$][attr] + gap * dif
    endfor
  newindex++
  $N = N - 1$
  endwhile
return //End of Populate//
```

Two parameters is needed in DBSCAN which are ε and MinPts. For each point in dataset D, the ε neighborhood of point p is shown in Equation (1). The ε neighborhood of each point in dataset D is processed. While the distance between p and q is calculated, Euclidean distance formula is used as given in Equation (2), where m is the dimension of the points. Then, the distance between any two points is determined and compared whether it is smaller than ε or not. The MinPts is used to check whether the point p is qualified to be the core point which means that the number of P' neighbors are greater or equal to MinPts [Equation (3)]. All density connected points form a cluster.

$$N_{\varepsilon}(P) = \{ q D | \text{dist}(p,q) \varepsilon \} \tag{1}$$

$$\text{dist}(p,q) = \sqrt{\sum_{i=1}^{m} (p_i - q_i)^2} \tag{2}$$

$$\left| N_{(p;\varepsilon)} \right| \geq \text{MinPts} \tag{3}$$

Density-based spatial clustering of applications with noise (Table 4) is applied in this work as the second layer process. After the comment classification is done, all comments in each class are further clustered in order to organize similar content into hierarchies for user.

3. FRAMEWORK

The framework of the comment management system (Fig. 2) consists of the feature extraction process, the classification task that applies SMOTE with Random Forest and the comment clustering. After the stopword removal and word stemming are completed, the feature selection is performed by chi-square test (χ^2) [Equation (4)]. The objective of using chi-square is to remove words that are unlikely to be significant. By calculating the chi-square scores for all words, we can rank the words by the chi-square scores, then choose the top ranked words for model training. This yield a dataset containing 800 word features that are further created as a feature vectors in the training set. Note that the higher value of the chi-square score, the more likelihood the feature is correlated with the class, thus it should be included for model training.

$$\chi^2(t,c) = \frac{\left(N * (AD - CB)^2\right)}{\left((A+C) * (B+D) * (A+B) * (C+D)\right)} \tag{4}$$

A = Number of comments in class c containing word t,

B = Number of comments in other classes (except c) containing word t,

C = Number of comments in class c that do not contain word t,

D = Number of comments in other classes (except c) that do not contain word t,

N = Total number of comments.

Given the feature vectors attached with the class label, the oversampling is performed on the minority classes using SMOTE. Then the ensemble learning algorithm, Random Forest starts learning from the training set. After the comment classification is finished, all comment in each class are further clustered in order to organize the comment into hierarchical structure (see Fig. 1). Note that comments that are classified as class Others are filtered out.

Table 4 | DBSCAN algorithm

Algorithm: DBSCAN

```
DBSCAN($D$, epsilon, min_points):
  $C$ = 0
  for each unvisited point $P$ in dataset
    mark $P$ as visited
    sphere_points = regionQuery($P$, epsilon)
    if sizeof(sphere_points) < min_points
      ignore $P$
    else
    $C$ = next cluster
    expandCluster($P$, sphere_points, $C$, epsilon, min_points)

expandCluster($P$, sphere_points, $C$, epsilon, min_points):
    add $P$ to cluster $C$
    for each point $P'$ in sphere_points
      if $P'$ is not visited
      mark $P'$ as visited
      sphere_points' = regionQuery($P'$, epsilon)
      if sizeof(sphere_points') ≥ min_points
      sphere_points = sphere_points joined with sphere_points'
      if $P'$ is not yet member of any cluster
      add $P'$ to cluster $C$

regionQuery($P$, epsilon):
    return all points within the $n$-dimensional sphere centered at $P$ with
      radius epsilon (including $P$)
```

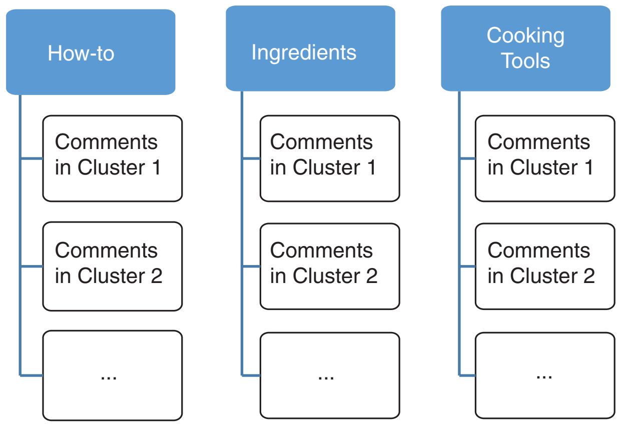


Figure 1 | Hierarchical structure obtained from the comment management system

Table 5 | Class distribution

Class	Number of instances
How-to	925
Ingredients	1,169
Cooking tools	301
Others	13,882

4. EXPERIMENTAL RESULTS

We retrieve comment posts on cooking video clip via website YouTube. There are 16,277 comments written in Thai language each of which is labeled into four classes. Class How-to consists of content regarding cooking methods or techniques whereas

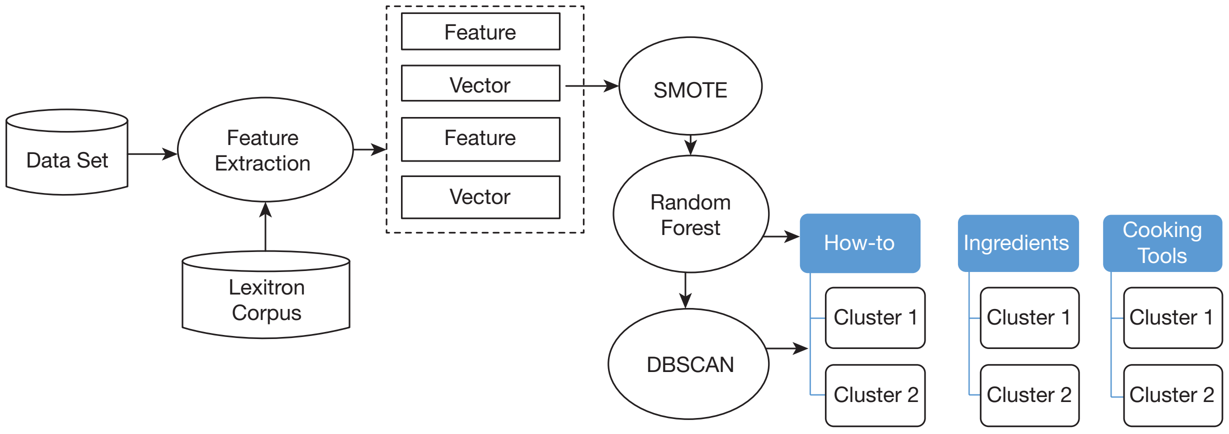


Figure 2 The comment management framework

comments of class Cooking tools mainly concern about cooking utensils. Class Ingredients focus on cooking recipes but class Others are miscellaneous contents such as admiring comment. Since Thai language has no word boundary, therefore Thai word segmentation tool namely Lexto that uses longest matching method is applied in the preprocessing step. The tool use Lexitron as a corpus. Stopword removal and word stemming are done as the preprocessing task.

To explore the learning algorithm's performance, two machine learning algorithm are evaluated, Random Forest, Naïve Bayes. The imbalanced problem is taken into consideration using synthetic minority class oversampling method namely SMOTE. We measure performance using precision, recall and F_1 (Equation 5–7), all experiments are performed based on 10-fold cross-validation.

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}} \tag{5}$$

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} \tag{6}$$

$$F_1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

The experimental result show that applying SMOTE with Random Forest provides the best performance compared to other algorithms (see Fig. 3 and Table 6). We found that F_1 score obtained from SMOTE + RF is 0.87 for Ingredients and Cooking tools class. For How-to class, F_1 score obtained from SMOTE + RF is 0.72 which is higher than that of SMOTE + NB and NB algorithm. We found that for imbalanced dataset using synthetic oversampling method with ensemble can enhance the classification performance. SMOTE + NB cannot improve the precision value for every class.

We found that NB performs better than SMOTE + NB since NB considers all features to calculate the conditional probability for each given class label. Although SMOTE algorithm increases the number of instances in minority classes but their feature values are synthesis to form the new instances. All of these feature values (a_i) are needed for conditional probability calculation that might affect classification of NB. Note that NB classify the instance using Equation (8).

$$V_{\text{NB}} = \arg\max P(C_j) * \prod_{k=1}^{n} P(a_i|c_j) \tag{8}$$

We know that Random Forest algorithm creates multiple learning models by randomly selects feature subset to create different decision trees and combines the final decision using majority vote.

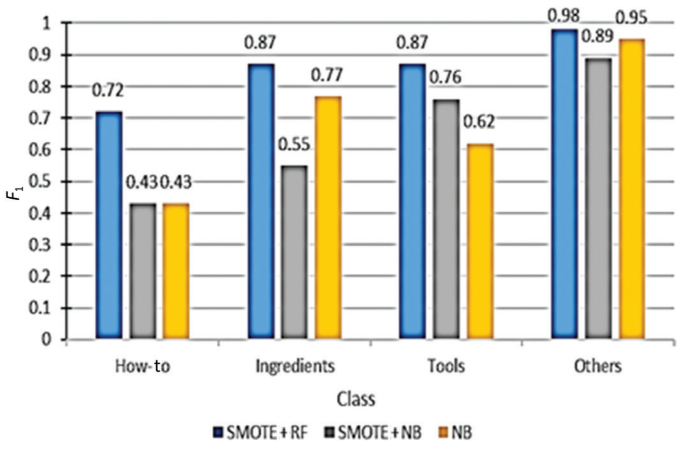


Figure 3 Average performance comparison

Table 6 Performance

Algorithm	Class	Precision	Recall	F_1
SMOTE + RF	How-to	**0.67**	**0.78**	**0.72**
	Ingredients	**0.83**	**0.91**	**0.87**
	Cooking tools	**0.91**	**0.84**	**0.87**
	Others	0.99	0.97	0.98
SMOTE + NB	How-to	0.33	0.66	0.43
	Ingredients	0.41	0.86	0.55
	Cooking tools	0.27	0.67	0.76
	Others	0.98	0.82	0.89
NB	How-to	0.35	0.56	0.43
	Ingredients	0.68	0.88	0.77
	Cooking tools	0.64	0.59	0.62
	Others	0.98	0.93	0.95

The advantage of ensembles with respect to single models show the robustness since there is no side effect on synthetic instances created by SMOTE.

5. APPLICATION

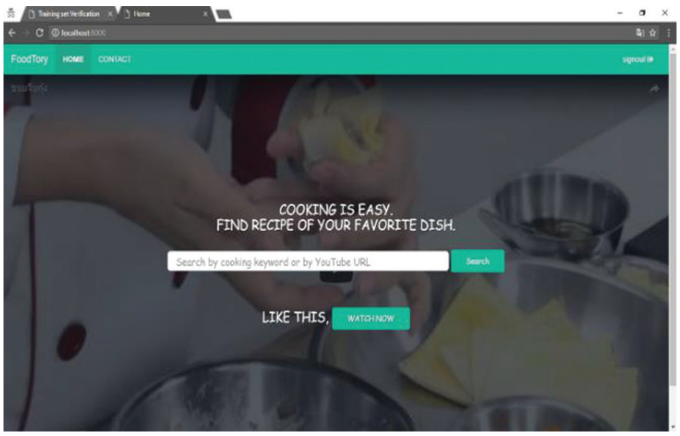
Our application has three main features as follows:

(1) Clip video feeder obtains user query to retrieve clip video and comments via YouTube API (see Figs. 4 and 5).

(2) Comment management system classifies all comments into four classes (see Fig. 6).

(3) The incremental learning facilitate administrator to update model's knowledge by adding new training set or relabel the misclassification instances (see Fig. 7).

Figure 7 shows the output of incremental learning feature that allows user to update new instances into the training set. User can do class labelling via the system and see the performance of the new model (see Fig. 8).



FoodTory
HOME
CONTACT
COOKING IS EASY.
FIND RECIPE OF YOUR FAVORITE DISH.
Search by cooking keyword or by YouTube URL
Search
LIKE THIS,
WATCH NOW

Figure 4 | Main user interface

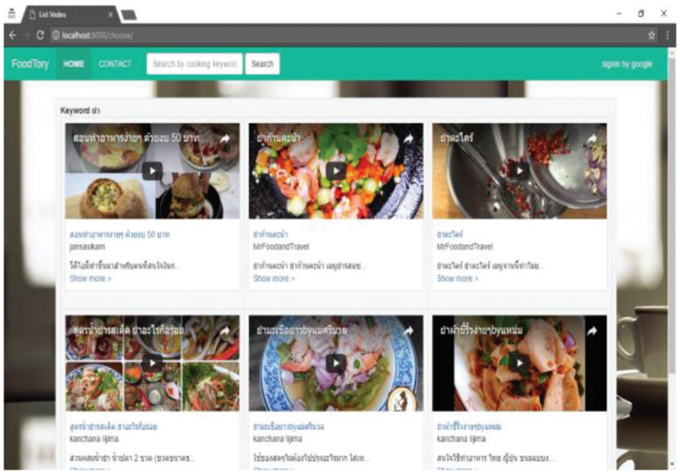
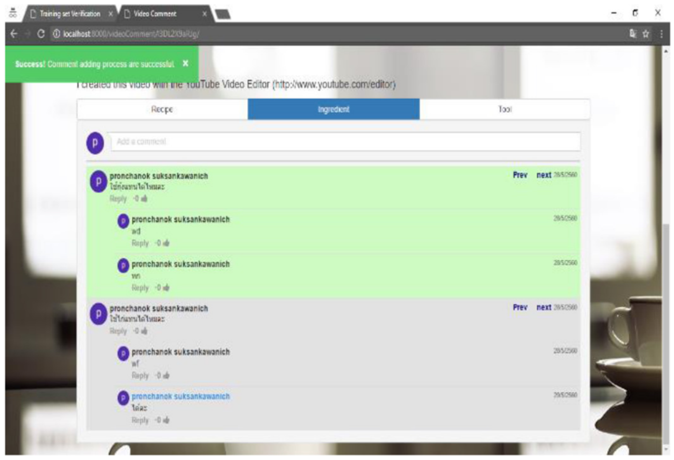
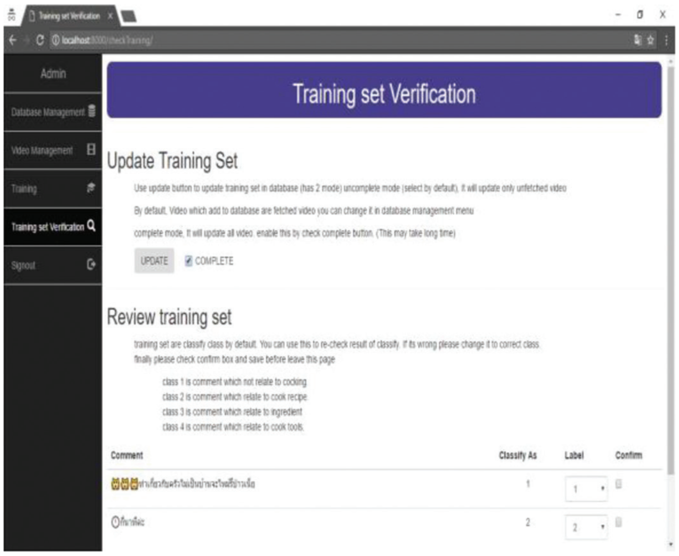


Figure 5 | Video Clip retrieval



Success! Comment adding process are successful
I created this video with the YouTube Video Editor (http://www.youtube.com/editor)
Recipe
Ingredient
Tool
Add a comment
pronchanok suksankawanich
Reply
Prev
next

Figure 6 | Comment management system organizes the posts into hierarchies



Training set Verification

Update Training Set

Use update button to update training set in database (has 2 mode) uncomplete mode (select by default), it will update only unfetched video

By default, Video which add to database are fetched video you can change it in database management menu

complete mode, It will update all video, enable this by check complete button. (This may take long time)

UPDATE [✓ COMPLETE]

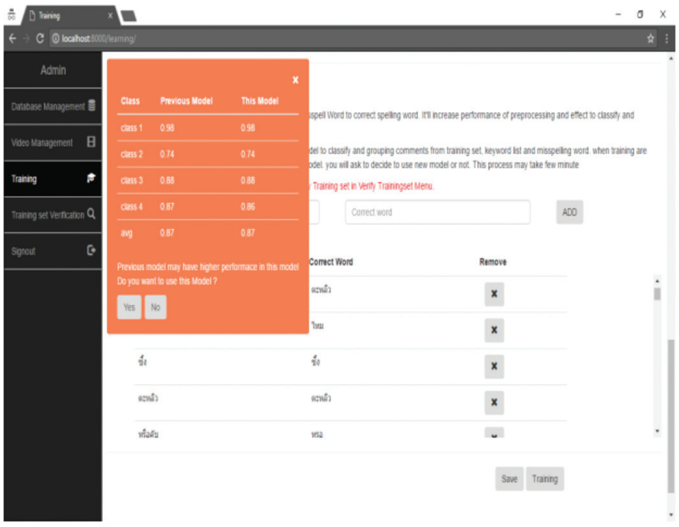
Review training set

training set are classify class by default. You can use this to re-check result of classify. If its wrong please change it to correct class. finally please check confirm box and save before leave this page

class 1 is comment which not relate to cooking
class 2 is comment which relate to cook recipe
class 3 is comment which relate to ingredient
class 4 is comment which relate to cook tools.

Comment	Classify As	Label	Confirm
[illegible]	1	1	☐
[illegible]	2	2	☐

Figure 7 | Training set updating facilities



Class	Previous Model	This Model
class 1	0.98	0.98
class 2	0.74	0.74
class 3	0.88	0.88
class 4	0.87	0.86
avg	0.87	0.87

Previous model may have higher performace in this model
Do you want to use this Model ?
Yes No

spell Word to correct spelling word. It'll increase performance of preprocessing and effect to classify and

del to classify and grouping comments from training set, keyword list and misspelling word. when training are odel. you will ask to decide to use new model or not. This process may take few minute

Training set in Verify Trainingset Menu.

Correct word ADD

	Correct Word	Remove
	[illegible]	x
	[illegible]	x
ซึ่ง	ซึ่ง	x
[illegible]	[illegible]	x
[illegible]	[illegible]	

Save Training

Figure 8 | The new model performance

6. CONCLUSION

In this work, we have studied and applied machine learning algorithms for the comment post in social media. The imbalanced problem is solved using SMOTE algorithm. There is no effect from minority class instances oversampling since the ensemble create multiple learning model using different feature sets.

The framework is implemented as the comment management system based on cooking video clips. The system provides options for the user to update the learning model in incremental style.

REFERENCES

[1] L. Breiman, Random Forests, Mach. Learn. 45 (2001), 5–32.

[2] T.G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization, Mach. Learn. 40 (2000), 139–157.

[3] Y. Freund, R. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1997), 119–139.

[4] P. Melville, R.J. Mooney, Creating diversity in ensembles using artificial data, J. Inf. Fusion. 6 (2004), 99–111.

[5] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002), 321–357.

[6] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96), Portland, Oregon, 2–4 August 1996, pp. 226–231.

[7] L. Qun, H. Xinyuan, Research on text clustering algorithms. In: Proceedings of the 2nd International Workshop on Database Technology and Applications, Wuhan, China, 2010, pp. 1–3.

[8] A.M. Bakr, N.M. Ghanem, M.A. Ismail, Efficient incremental density-based algorithm for clustering large datasets, Alexandria Eng. J. 54 (2015), 1147–1154.