Effect of Synthetic Minority Oversampling Technique (SMOTE), Feature Representation, and Classification Algorithm on Imbalanced Sentiment Analysis

Widi Satriaji
Department of Informatics
Universitas Diponegoro
Semarang, Indonesia
satriajiwidi@student.undip.ac.id

Retno Kusumaningrum
Department of Informatics
Universitas Diponegoro
Semarang, Indonesia
retno@live.undip.ac.id

***Abstract*— The comments received on Internet-based online hotel reservation services are an important resource that can be utilised by hotel service providers including hotel managers' and hoteliers' for exercising quality control measures in their hotel reservation service. Importantly this contributes towards increased customer satisfaction and hotel revisits. In this study, Sentiment Analysis (SA) is used to analyse the comments received from customers. However, there are several problems associated with SA such as the unequal number of each class of data (imbalanced datasets), the classification algorithm and the feature representation. Using SMOTE (Synthetic Minority Oversampling Technique) this research aims to investigate how this technique balances the amount of data from each class employing; the Naïve Bayes (NB), Logistic Regression (LR), and Support Vector Machine (SVM) classification algorithms. And also using; term presence (TO), term occurrence (TO), and Term Frequency-Inverse Document Frequency (TF-IDF) feature representations to gauge the effect on the performance of sentiment analysis. The findings from the study found that the use of SMOTE was effective in improving the model's classification performance when data is imbalanced, as evidenced by the average model performance improvement of approximately 12 %. Furthermore, feature representation of TO resulted in an average of 81.68 % of the G-mean Score, followed by TP of 79.89 %, and TF-IDF 79.31 %. As for the classification algorithm, LR resulted in an average score of 81.65 % of the g-mean score, followed by SVM 81.55 %, and NB of 77.68 %.**

Keywords*— *Sentiment analysis, hotel, imbalanced datasets, SMOTE, g-mean score.

I. Introduction

The online hotel booking industry globally has become recognised thanks to the proliferation of Internet-based devices, especially in Indonesia. In fact, many of these services are evident in both web-based and smartphone application development. Most service providers in this industry provide comment or note fields for users to express their feedback (i.e. impressions and criticisms) regarding the quality they receive from hotels in which they register and make a booking. These comments are an important resource which is used to provide feedback to respective hotels, and particularly for hotel managers and hoteliers to exercise quality control measures for hotel booking services resulting in increased customer satisfaction and revisits.

Sentiment Analysis (SA) is a tool that can be utilised to analyse hotel booking comments. SA is a research area that is used to analyse the expression of opinions, e.g. from the World Wide Web (WWW), also called the web [1]. Cambria et al. [2] classify SA techniques into four main categories, namely; keyword-spotting, lexical affinity, concept-based, as well as statistic and machine learning. The machine learning (ML) technique is the method employed in this research.

Some of the ML algorithms that are often used and have been proven optimal for SA are Naive Bayes, Logistic Regression, and Support Vector Machine. However, no single research has provided extensive generalisation claims for either. Therefore, in this research the performance of the three algorithms will be compared mentioned previously solve one problem, namely performing SA on the comments on a hotel booking website.

However, the problem that arises later is that most of these comments tend to be rather imbalanced regarding the number of each class or incline to one of the poles, for example, to the negative or vice versa. In general, ML algorithms for classification will result in a model with minimal sensitivity to minority classes when receiving imbalanced datasets [3]. Furthermore, it has proven to be quite a challenging problem for the ML research community [4] as it will undoubtedly lead to bad performance regarding the SA to be performed.

Some of the approaches proposed to address the problem of distribution imbalances in the dataset include re-sampling [5], one-class classification [6], and cost-sensitive learning [7]. The approach adopted in this research is the re-sampling technique.

SMOTE (Synthetic Minority Over-sampling Technique) is one of the most commonly used over-sampling techniques used to handle imbalanced datasets by creating synthetic data in minority data classes. This is so that the data becomes balanced [5] and is expected to aid in better classification performance [8].

The third problem that often arises in SA is about choosing the type of feature vector representation. The representation of features to be used in this case are term presence, term occurrence, and term frequency-inverse document frequency (TF-IDF).

The performance metrics that are proposed to use in this research are G-mean score/geometric mean score [4][9]. This metric represents a good performance balance for both the major and minor classes, which therefore means that poor performance in minority classes will result in poor G-mean

978-1-5386-7440-6/18/$31.00 ©2018 IEEE

values even though the performance of the majority class is excellent [10].

The rest of this paper is organized as follows. Section 2 describes the implemented methodology. The experimental scenarios and results are described in Section 3. Subsequently, conclusions are drawn in Section 4.

II. METHODOLOGY

This section will discuss the research methodology conducted. Fig. 1 describes the methodology that is used in this study.

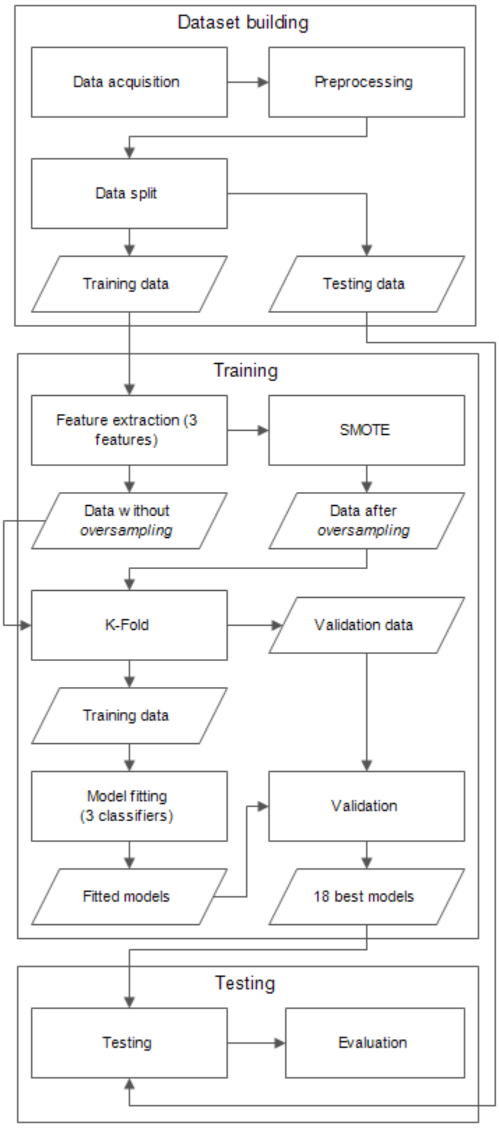

Fig. 1. Overview of this Research

TABLE I. EXAMPLES DATASET.

Text	Class
Pelayanan yang ramah dan keadaaan hotel yang bersih	pos
Banyak jentik nyamuk, sempat pindah kamar, ternyata setelah pindah kamar, sama saja banyak jentik nyamuk	neg

A. Data Acquisition

Data collection in this research uses hotel comment data obtained from the Traveloka website using manual scraping without using an application programming interface (API). From the scraping process, approximately 13,000 comments [data] are collected from all locations/cities available from the site. Next, 1,500 comments (data) are taken at random which are categorised/grouped manually into two distinct classes (ground truth); whether the comments are positive or negative. A ratio is then formulated of the number of positive and negative data with a value of 1058:442. The data will then be used as training data to form the SA classification model.

B. Pre-processing

Pre-processing is performed with the aim to prepare raw data obtained from the acquisition process with scraping into data in preparation for the next process. Pre-processing consists of several sub-processes, namely: tokenisation, stopwords removal, stemming, and normalisation. The tokenisation is performed in order to form comment data that is initially, a sentence or paragraph, into separate words [11]. The words are then stored into arrays after normalisation using the Jaro-Winkler Distance algorithm to correct any typographical errors (typos) [12]. From the array of words, the words that are then considered less representative are omitted or commonly referred to as stopwords [13]. This process is called stopwords removal. After that, the next process that follows is the stemming process; namely simplification of the word to be the basic word to eliminate dimension space of the data [14].

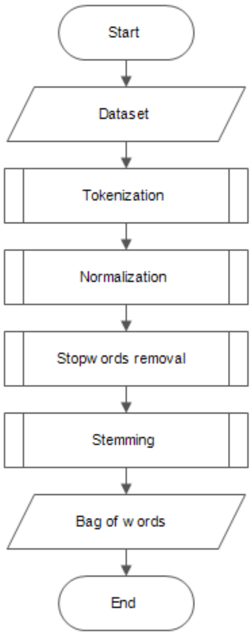

Fig. 2. Preprocessing Steps

TABLE II. EXAMPLES OF PRE-PROCESSED DATASET.

Text	Class
['layan', 'ramah', 'ada', 'hotel', 'bersih']	pos
['banyak', 'jentik', 'nyamuk', 'pindah', 'kamar', 'pindah', 'kamar', 'sama', 'banyak', 'jentik', 'nyamuk']	neg

C. Data Split

The process of split data is undertaken with the aim to separate the data, in order to obtain the training data and testing data. From the 1,500 data, having a ratio of 1058:442

for the data with positive and negative labels, 158 positive data and 142 negative data were collected for the testing data. This results in 300 test data and the remaining 1,200 as training data. Before splitting the data, random data sequencing/shuffle is conducted so that the results of the training and testing data are random in sequence. Accordingly, this is undertaken to minimise the bias on the performance measurement of the classification model that will be undertaken later.

D. Feature Extraction

The feature extraction process is divided into 2 sub-processes; sub-process of vocabulary formation and sub-process of document vectorisation into a feature vector. Vocabulary formation is performed as a prerequisite for vectorisation, in which the vector dimension is equal to the length of the vocabulary. In this case, the order of values in each vector is the value of the word in the corresponding document. Accordingly, there are 3 types of representation of the feature vector to be formed, namely:

- *Term presence:* the presence or absence of each word in the vocabulary for each word of each comment;

  extracted features example:

  [1, 0, 1, 1, 0, 1, 1, 0]

- *Term occurrence:* how often each word appears in the vocabulary for each word of each comment; and

  extracted features example:

  [2, 0, 2, 2, 0, 2, 2, 0]

- *TF-IDF:* re-weighting results at corpus level to term occurrence [15]. Referred to as 'TF'.

  extracted features example:

  [0.03, 0.00, 0.07, 0.04, 0.00, 0.08, 0.01, 0.00]

E. SMOTE

SMOTE is an algorithm for the oversample of minor class data [5]. Simply put, SMOTE performs oversampling by taking k data from k-NN for each data in the minor class. Then for N/100 number of magnifications, generate new data for each minor class data by retrieving "in line" data with one of the randomly selected k-NN data results.

The (simplified) steps of the SMOTE algorithm are as follows:

1. Select $\boldsymbol{k}$ number of nearest neighbours to use and how many times $\boldsymbol{l}$ = N / 100 oversamples will be performed.
2. For each of the minority class x data points, do:
3. Select randomly $\boldsymbol{l}$ data from the nearest neighbour's data $\boldsymbol{k}$.
4. For each of these data, form synthetic data by taking the random data points that are aligned with the present x points with each observed $\boldsymbol{l}$.
5. The return of the SMOTE algorithm is the initial data plus the oversampling results that have been performed.

The magnification undertaken at SMOTE in this research is at 200 %, which means to enlarge the data of the minor class into 3-fold the original amount or to increase the result of generating oversampling data by 2-fold the initial size. The ratio of the original training data amount before the oversampling of the positive and negative classes is 900:300, which following the oversampling of SMOTE becomes balanced by a ratio of 900:900 for each class.

F. Classification Algorithms

Some of the classification algorithms used in this research are; Naïve Bayes (NB), Logistic Regression (LR), and Support Vector Machine (SVM).

- Naïve Bayes is renowned for being fast, inexpensive computing, accurate, and reliable. In addition to being successful in many cases, this algorithm is also known to work optimally and popularly in the field of Natural Language Processing especially in text classification [16].
- Logistic regression algorithm is an actual classification algorithm derived from a linear regression algorithm, but with a non-linear function applied to the output outcome. Non-linear functions are a logistic function or sigmoid function. The sigmoid function itself is a function that produces a curve with a character set such as the letter S (S-shaped curve) [17].
- The idea behind the Support Vector Machine is to find the best hyperplane that separates the datasets into each class. In a two-dimensional space, a hyperplane is a line that divides the dataset into its own class. That means the hyperplane is a separator between dataset classes with n-1 dimensions of the dataset dimension [18].

G. Evaluation

Performance metrics are used to perform the performance measurement results from the model that have been formed. Moreover, the performance metric to be used in this research is g-mean which is calculated as the product of prediction accuracy for both classes. Even though a model classifies most data correctly, poor performance in predicting minority instances will result in a low average g-mean. Therefore, this performance metric is suitable for use in studies with imbalanced datasets. In fact, g-mean is important enough to measure overfitting for the majority class and the extent to which minority classes are ignored [10].

$$G-mean = \sqrt{\frac{TN}{TN+FP} \times \frac{TP}{TP+FN}} \quad (1)$$

In this research, where the amount of positives data are more dominant compared to the negatives, TP/true positives score will be high and TN/true negatives will be the other way around. With SMOTE, better score of TN (true negative rate) of the model is expected.

III. EXPERIMENTAL RESULT

A. Training Evaluation

In the first scenario we will see the performance based on the g-mean score on the training process of each combination which is a total of 18 combinations obtained from the combination of 3 classification algorithms: NB, SVM, and LR, 3 feature representations: term presence (TP), term occurrence (TO), TF-IDF (TF), and whether (Y) or not (T) oversampling (SMOTE) is applied. The value of the g-mean score in this scenario is the average of the result of the g-mean score of the training process of the 10-fold on the 10-fold cross validation that has been performed.

The table above shows a very distinct pattern indicating an increase in model performance on all combinations of the training process after oversampling (SMOTE) in minority class data. Accordingly, an average performance increase of approximately 16 % occurs after oversampling. This indicates that the oversampling process with SMOTE is quite effective in improving model performance in the case of the classification with imbalanced data.

TABLE III. THE RESULT OF SCENARIO I OF 18 COMBINATIONS.

Combination	Algorithm	Feature	Sampling	G-mean Score (%)
K1	NB	TP	T	78.59
K2	NB	TO	T	80.17
K3	NB	TF	T	61.92
K4	SVM	TP	T	76.90
K5	SVM	TO	T	79.48
K6	SVM	TF	T	78.61
K7	LR	TP	T	76.97
K8	LR	TO	T	76.64
K9	LR	TF	T	68.79
K10	NB	TP	Y	89.64
K11	NB	TO	Y	89.65
K12	NB	TF	Y	90.28
K13	SVM	TP	Y	92.38
K14	SVM	TO	Y	93.03
K15	SVM	TF	Y	93.44
K16	LR	TP	Y	91.94
K17	LR	TO	Y	92.42
K18	LR	TF	Y	91.25

In the NB classification algorithm, there is an increase in performance when the oversampling is undertaken because the minority oversampling process improves the individual likelihood of the words (features) that should be more suitable or stronger into the minority class. Whereas, on the LR classification algorithm, oversampling improves the cost function calculation, which increases the frequency of errors when minor class data is classified. Lastly, the SVM classification algorithm in which the oversampling improves performance by updating the support vectors used in forming the dividing hyperplane that ultimately shifts the hyperplane away from the minority class region.

In the comparison of training performance based on the representation of features used, it is generally seen from the average that the best performance resulted from the representation of the TO type feature with the value of 85.23 %, followed by the TP of 84.40 % and finally, TF of 80.72 %. Indeed, the performance generated by TO and TP is relatively similar to most cases, and TO is better than TP because TP loses information regarding how often a term appears in the document. Although, in the comment data with a relatively short length, the difference from the same word occurrence frequency tends to be minimal. Notwithstanding, the performance gaps may be more significant in longer-term data types. As for TF, especially on the NB algorithm before oversampling, the performance difference compared to the other types of features is relatively far away. This is due to the distribution/continuous value of the TF features being relatively broader.

B. *Testing Evaluation*

In the second scenario the performance of the classification model is seen based on the g-mean score on the testing process of each combination of 18 combinations obtained from the combination of 3 classification algorithms: NB, SVM and LR, 3 representation features: TP, TO, TF, and whether (Y) or not (T) oversampling (SMOTE) is applied.

In this scenario, there is an average increase of approximately 12 % after oversampling. The difference in the magnitude of the increase in the testing process compared to the training process (12 % versus 16 %) is due to some unseen data on the comment dataset in the testing process resulting from the data splitting process.

In the comparison of training performance based on the representation of features used, it is generally seen from the average that the best performance resulted from the representation of the feature type TO with an average value of 81.68 %, followed by the TP of 79.89 % and lastly, TF of 79.31 %. The pattern is like the first scenario outlined previously. Therefore, this shows that the TO feature is best suited for SA based on the average performance of the training and testing process in both the first and seconds scenario undertaken before.

TABLE IV. THE RESULT OF SCENARIO II OF 18 COMBINATIONS.

Combination	Algorithm	Feature	Sampling	G-mean Score (%)
K1	NB	TP	T	69.81
K2	NB	TO	T	70.53
K3	NB	TF	T	62.40
K4	SVM	TP	T	76.70
K5	SVM	TO	T	79.38
K6	SVM	TF	T	78.43
K7	LR	TP	T	76.88
K8	LR	TO	T	79.99
K9	LR	TF	T	72.46
K10	NB	TP	Y	86.96
K11	NB	TO	Y	88.43
K12	NB	TF	Y	87.92
K13	SVM	TP	Y	82.71
K14	SVM	TO	Y	85.47
K15	SVM	TF	Y	86.63
K16	LR	TP	Y	86.29
K17	LR	TO	Y	86.29
K18	LR	TF	Y	88.00

The results of TO and TP are relatively compareable in several cases, but TO is mostly better than TP because TP lost the information about how frequent a term appears in a document, despite the fact that in comments-formed dataset with count of words that relatively small, the appearance frequency from the same term is quite minimal. Performance gap maybe broader in the type of data that has more words.

The best classification algorithm performance seen from the average is LR with an average value equal to 81.65 %, SVM equal to 81.5 5%, and NB equal to 77.68 %. Notably, in this case, it is interesting to note that in the testing process LR is the best, replacing SVM. The average is obtained because LR produces an average increase in performance which is greater than SVM after the oversampling process.

The further interesting point is that although the average NB is the most unfavourable compared to the other two, the NB produces the best unit performance on the combination with the TO feature after oversampling with a value of 88.43 %. NB also generates the largest average for performance after oversampling.

C. Training vs. Testing

In the third scenario a performance comparison will be seen based on the g-mean score between the training and the testing process of each combination of 18 combinations obtained from the combination of 3 classification algorithms: NB, SVM and LR, 3 representation features: TP, TO, TF, and whether (Y) or not (T) oversampling (SMOTE) is applied

The average training performance is 83.4 %, while the average performance of testing is 80.2 %. From the average, it can be seen that the difference in performance of about 3 % is due to model overfitting in the training process. Indeed, this may be justified, given that the performance measurement of training (validation) is carried out using the same data with the data used in forming the model, in which all data (vocabulary feature) have been seen. While in the testing process, in order to be as close as possible to imitate the real-world situation, not all the features have been seen/represented in the previous training process. So, the words that should probably be important are discarded because they do not belong to the vocabulary that has been formed.

Before oversampling, the performance of each model and combination with features is relatively balanced in the training and testing process, or sometimes it improves in the testing process as in the case of K8 and K9. In the case of the NB algorithm, there is a considerable decline in testing performance, especially for the TP and TO features. However, after oversampling, the NB displays its strength by generating the least difference in the performance of training to testing, compared to the two other algorithms. Even though the training performance in NB is not better compared to the others, its testing performance is the best. Therefore, this concludes that NB has the best generalisation ability among the three.

IV. Conclusions

Based on the analysis and testing results in this research, several conclusions are presented as follows:

1. The use of SMOTE is quite effective (based on the g-mean score) towards improving model performance in the case of the classification with imbalanced dataset, as evidenced by the average model performance improvement of approximately 12 %.
2. Feature representations that produce the best classification performance of the model, based on the average value, are TO with G-mean score of 81.68 %, followed by the TP of 79.89 %, and TF of 79.31 %.
3. The classification algorithms that produce the best classification performance of the model, based on the average value are LR with the G-mean score equal to 81.65 %, followed by SVM equal to 81.55 %, and NB equal to 77.68 %.

References

[1] A. Mountassir, H. Benbrahim and I. Berrada, "An empirical study to address the problem of Unbalanced Data Sets in Sentiment Classification," Seoul, 2012.

[2] E. Cambria, B. Schuller, Y. Xia and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intell Syst,* 2013.

[3] H. He and Y. Ma, Imbalanced Learning: Foundations, Algorithms, and Applications, Wiley-IEEE Press, 2013.

[4] M. Kubat and S. Matwin, "Addressing the Curse of Imbalanced Training Sets: One-Sided Selection," *Department of Computer Science University of Ottawa,* 1997.

[5] N. Chawla, K. Bowyer, L. Hall and W. Kegelmeyer, "SMOTE: Synthetic Minority Over-Sampling Technique," *Journal of Artificial Intelligence Research,* vol. 16, p. 321–357, 2002.

[6] P. Juszczak and R. Duin, "Uncertainty Sampling Methods for One-Class Classifiers," Washington DC, 2003.

[7] Z. Zhou and X. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Transaction on Knowledge and Data Engineering,* vol. 18, no. 1, pp. 63-77, 2006.

[8] J. Ah-Pine and E. P. S. Morales, "A Study of Synthetic Oversampling for Twitter Imbalanced Sentiment Analysis," *DMNLP,* 2016.

[9] M. Bekkar, H. K. Djemaa and T. A. Alitouche, "Evaluation Measures for Models Assessment over Imbalanced Data Sets," *Information Engineering and Applications,* vol. 3, no. 10, p. 27, 2013.

[10] H. Shohei, K. Hisashi and T. Yutaka, "Roughly Balanced Bagging for Imbalanced Data," *Statistical Analysis and Data Mining,* vol. 2, no. 5-6, pp. 412-426, 2009.

[11] D. S. Kannan and V. Gurusamy, "Preprocessing Techniques for Text Mining," *International Journal of Computer Science & Communication Networks,* vol. 5(1), pp. 7-16, 2014.

[12] Y. Rochmawati and R. Kusumaningrum, "Comparison Study of String Searching Algorithms in Approximate String Matching Method for Identifying Text Typing Errors," *Jurnal Buana Informatika,* vol. 7, no. 2, pp. 125-134, 2016.

[13] M. F. Porter, "An Algorithm for Suffix Stripping," *Program,* vol. 14, no. 3, pp. 130-137, 1980.

[14] C. Ramasubramanian and R. Ramya, "Effective Pre-Processing Activities in Text Mining using Improved Porter"s Stemming Algorithm," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 2, no. 12, pp. 4536-4538, 2013.

[15] V. John, "A Survey of Neural Network Techniques for Feature Extraction from Text," *arXiv,* 2017.

[16] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," *AAAI Workshop,* pp. 41-48, 1998.

[17] C.-Y. J. Peng, K. L. Lee and G. M. Ingersoll, "An Introduction to Logistic Regression Analysis and Reporting," *The Journal of Educational Research,* vol. 96(No. 1), pp. 3-14, 2002.

[18] S. R. Gunn, Support Vector Machines for Classification and Regression, Southampton: University of Southampton, 1998.