Dealing with Class Imbalance in Sentiment Analysis using Deep Learning and SMOTE

Shweta Kedas *[1], Arun Kumar †[1], and Puneet Kumar Jain ‡[1]

[1]Department of Computer Science and Engineering, National Institute of Technology, Rourkela, Odisha, India - 769008

Abstract In textual data, sentiments or opinions expressing polarities (positive or negative) often form the basis for human decision-making. Therefore, sentiment analysis has always been an important area of research in the field of artificial intelligence. Recently, deep learning models have been used for the sentiment analysis. However, the class imbalance of the dataset adversely affects the performance of these models. To address this issue, the paper presents a method to resample the dataset using Synthetic Minority Over-sampling Technique (SMOTE). The proposed method is applied to three different datasets of customer reviews, each of which exhibits different class imbalance ratios. To show the impact of the method, the modern Recurrent Neural Network (RNN) based architectures (LSTM, GRU, and Bi-directional) are trained with both the originally imbalanced datasets and the SMOTE balanced datasets, and comprehensively analysed the performance of these approaches. The obtained results show that the models trained with balanced datasets outperform other methods across most models with significant improvements over the original dataset.

1 Introduction

The procedure of capturing patterns and understanding people's opinions from data collected through questionnaires had first taken place in the early 20th century. This work laid the foundation for a branch of computational linguistics called Sentiment Analysis. In the literature, there are two main approaches for the sentiment analysis including lexicon-based approach and machine learning-based approach [2, 3, 9, 11, 18]. Lexicon-based approaches are used to identify the words' polarity and than the frequency-based and location-based approaches have been used to classify the text [1]. Machine learning methods are being used extensively for the sentiment analysis due to availability of adequate dataset and advancement of learning techniques [10, 21, 29]. In recent times, deep learning based machine learning

* shweta.kedas@gmail.com

† kumararun@nitrkl.ac.in

‡ jainp@nitrkl.ac.in

methods have shown to out-perform the classical machine-learning methods [9, 11]. However, performance of the machine learning models significantly get affected due to the class imbalance in the dataset [13]. A dataset is said to be imbalanced if imbalance ratio (IR) > 1.5 [16]. IR is the ratio of cardinality of the majority to the minority class. Imbalance in class impacts the accuracy of the analysis method because when a machine learning method will be trained with the imbalance dataset, the trained model will be biased towards the majority class [10]. Therefore, machine-learning models have to adopt approaches to deal with class imbalance.

Cost-sensitive learning methods, data-level preprocessing methods, and algorithm-level methods are three major approaches to deal with the class imbalance issue [4, 10, 14, 16]. Prustoa et al. [24] used random undersampling (RUS) for Twitter sentiment classification using machine learning techniques. On the other hand, Ah-Pine et al. [1] implemented oversampling techniques SMOTE [6], Borderline-SMOTE, and ADASYN methods on imbalanced Twitter datasets using decision trees and logistic regression. To balance the image dataset, Dablain et. al. proposed a deep learning and SMOTE based approach and termed it as DeepSMOTE [26]. The proposed approach is consists of three major components: (i) an encoder/decoder framework; (ii) SMOTE-based oversampling; and (iii) a dedicated loss function that is enhanced with a penalty term.

In the literature, various studies have addressed class imbalance issue, although impact of these approaches on deep learning approach for sentiment analysis has to be explored. Therefore, the main objective of the presented work is to use the SMOTE algorithm and to analyse the impact of it on the sentiment analysis performed using the deep learning models. The original class imbalanced dataset and dataset blanaced using SMOTE is applied to the long Short Term Memory (LSTM) [12] and Gated Recurrent Units (GRU) [7] models for sentiment classification.

The rest of the paper is structured as follows: Section 2 presents the methods and material including the dataset used for experiment, the SMOTE algorithm, and the deep learning models. Section 3 presents and discusses the results of the experiments. Finally, section 4 concludes the work with a discussion on the potential directions for future research.

2 Methods and materials

The block diagram of the proposed work is depicted in figure 1.1. Three main components of the block diagram, dataset, SMOTE algorithm, and RNN models are described in the following subsections.

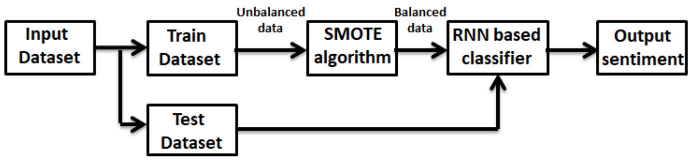

Fig. 1.1 Block Diagram of the SMOTE based sentiment classification

2.1 Dataset

The dataset is collected from the Amazon which consists of reviews belong to the three categories- software, fine foods, and appliances and abbreviated as D1, D2, and D3 respectively in this study [20]. Total number of reviews in D1, D2, and D3 is 459436, 568454, and 602777 respectively. Each review is consists of the ”ProductId”, ”UserId”, ”ProfileName”, and ”overall”. Duplicate entries are removed from the dataset based on the combination of the columns ”ProductId”, ”UserId” and ”ProfileName”. The column ”overall” describes the rating given by the users with 5 being excellent and 1 being bad. Since the work deals with binary sentiment classification, the ratings 1 and 2 are mapped to 0 (negative), 4 and 5 are mapped to 1 (positive) and rating 3 is not considered as it exhibit neither positive nor negative sentiment. Figure 1.2 depicts the number of positive and negative reviews, and the IR in each dataset post data cleaning.

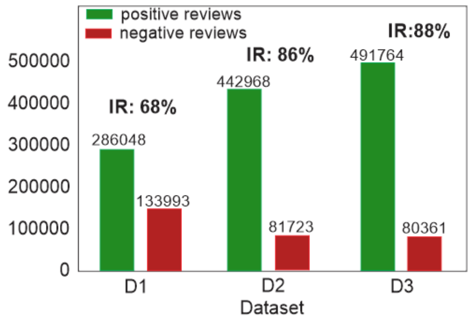

Fig. 1.2 Sentiment distribution of the reviews for each dataset

2.2 SMOTE Algorithm

SMOTE is an oversampling technique used to create synthetic samples of the minority class [6]. It is an iterative approach which considers the k-nearest neighbor (default k = 5) samples belong to the minority class, and uses random interpolation to compute synthetic samples. This algorithm focuses on the feature space and creates synthetic data points as a convex combination of the chosen minority class data points, and it's nearest neighbors.

Figure 1.3 illustrates an example of SMOTE generated synthetic sample in 2-dimensional feature vector space. The easier it is to generate SMOTE-based synthetic samples for a given class, clearer the boundary between the word vectors belonging to positive and negative classes. The time taken for SMOTE algorithm to produce synthetic samples for dataset D1, D2, and D3 is respectively 30 min 8 sec, 8 min 37 sec, and 11 min 20 sec respectively. The time taken for dataset D1 is much higher than the other datasets. This might be due to the less imbalance ratio, implying the overlapping of positive and negative class data points in vector space. While in datasets D2 and D3, SMOTE algorithm consumed less time which implies that the positive and negative class samples are clearly differentiated and grouped.

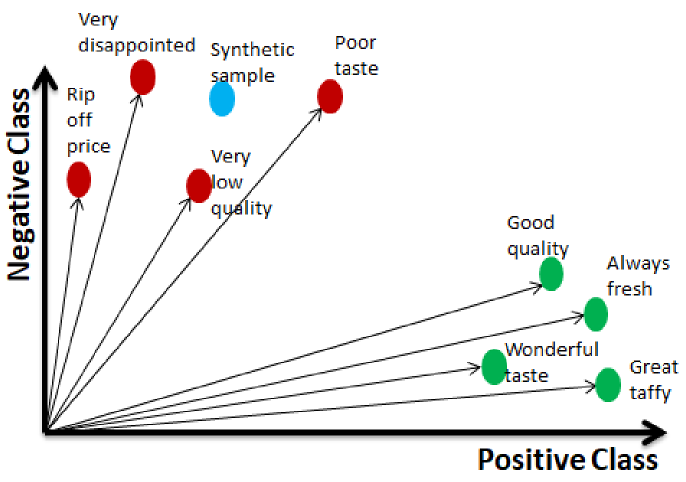

Fig. 1.3 Illustration of SMOTE generated synthetic samples using word vectors belonging to different classes

2.3 Deep learning Models

In this work, three RNN based models including LSTM, GRU, Bi-directional LSTM (BiLSTM), and Bi-directional GRU (BiGRU) models are implemented using Keras library [8].

Long-Short Term Memory (LSTM) [12] is a type of RNN which consists of input gate, output gate, and forget gate, which is inspired from logic gates to control the information flow within the LSTM units and memory cells.

Gated Recurrent Units (GRU) [7] is similar to LSTM except it consists of only two gates, reset and update gates, and memory cells. The gates of GRU slightly restrict the flow of information by updating to learn necessary information based on the importance of the sequential data and skip unnecessary information.

Bi-directional RNNs (Bi-RNN) [27]: The Bi-RNN working technique is similar to the look-back and look-ahead ability of Hidden Markov Models [30]. The Bi-RNNs consist of two hidden layers, one each for forward and backward passes.

These models are trained with the original imbalanced data and SMOTE balanced data. Before training, the text sequences are cleaned by removing punctuation and HTML tags using regular expressions. Next, the Natural Language Processing techniques: Stemming and Lemmatization are performed on the text. Then the Keras Embedding layer is used to encode the input using Keras Tokenizer API. The advantages of using the embedding layer are as follows:

1. The embedding layer derives word embeddings from the input document, and this layer can be saved and used in other models.
2. The embedding layer is flexible and can be utilized to load pre-trained word vectors.
3. Being implemented as a part of the neural network model, the embedding can comprehend from the model itself.

The hyperparameters of the proposed models are provided in Table 1.1. To measure the proposed models' performance, the logarithmic loss function (binary cross-entropy) [17] is used. The models are optimized using 'RMSProp' with an initial learning rate of 0.001, since RMSProp converges faster compared to other optimizers [25]. To overcome the overfitting of the models', dropout and early stopping regularization methods have been implemented [28]. In this work, a dropout layer is added between the RNN layer and a fully connected dense layer with a 0.3 probability. To overcome the issue of overfitting, Early stopping [23] regularization method is used that halt the training before the model cam overfit. In early stopping, to decide the stopping point for model training, we choose "validation loss" with mode set to "minimum" for early stopping, as lowering the loss value improves the model's performance.

Table 1.1 Hyperparameters of the models

Parametre	**Value**
Input Length	100
Batch size	128
Dropout	0.3
Loss function	Logarithmic loss
Optimizer	RMS Prop

3 Results and discussion

To evaluate the models' performance, we use the metrics F1-score and area under curve (AUC) of receiver operating characteristic (ROC) curve [4]. The accuracy along with F1 scores and AUC trained on the three datasets in Table 1.2. Also, the precision and recall values for each class for every dataset are compared. F1-score, a measure of test accuracy, which gives equal weightage to precision and recall, makes it an appropriate performance metric. ROC metric is only used for binary classification problems. The ROC curve is a plot of sensitivity (Y-axis) and inverted specificity (X-axis) for different threshold values, ranging from 0 to 1. The area under curve summarizes the model's ability to classify. High AUC for a given binary classification model implies that the model can correctly distinguish between the positive and negative classes. Figures 1.4, 1.5 and 1.6 visualizes the precision and recall scores of each class for datasets D1, D2 and D3 respectively.

Table 1.2 Results of our models across datasets and data distribution

		D1			D2			D3		
Data Distribution	**Model**	**Accuracy (%)**	**F1 Score (%)**	**AUC**	**Accuracy (%)**	**F1 Score (%)**	**AUC**	**Accuracy (%)**	**F1 Score (%)**	**AUC**
Original	GRU	92.57	89.30	0.9571	96.77	94.11	0.9698	93.41	93.26	0.9694
	LSTM	93.84	89.35	0.9558	95.83	94.44	0.9691	94.43	93.45	0.9711
	Bi-GRU	93.52	89.13	0.9556	94.69	92.85	0.9612	94.59	92.84	0.9660
	Bi-LSTM	93.54	89.20	0.9562	92.80	91.59	0.9551	96.13	92.79	0.9677
SMOTE	GRU	91.03	89.33	0.9578	96.00	94.60	0.9686	96.27	**94.01**	**0.9743**
	LSTM	92.21	89.57	0.9584	97.44	94.42	0.9688	95.85	93.60	0.9730
	BiGRU	93.17	89.39	0.9568	95.87	94.30	0.9685	96.45	93.86	0.9723
	BiLSTM	91.93	**89.68**	**0.9585**	97.21	**94.61**	**0.9702**	95.68	93.96	0.9731

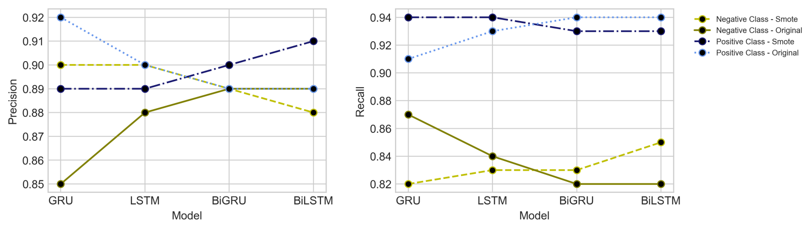

Fig. 1.4 Precision and Recall values of models trained with dataset D1

From Table 1.2, it can be observes that for the dataset D1, the F1-score and AUC of all the models are slightly high in case of the SMOTE generated dataset. The precision and recall values for each class are balanced in case of both original and SMOTE generated datasets of D1. Overall, it can be concluded that models trained with SMOTE generated dataset performed better than the original dataset. The precision and recall scores of models trained with dataset D1 are shown in figure 1.4.

The model results for dataset D2 from table 1.2 show that all the three metrics; accuracy, F1 and AUC values are slightly high for SMOTE dataset. From figure

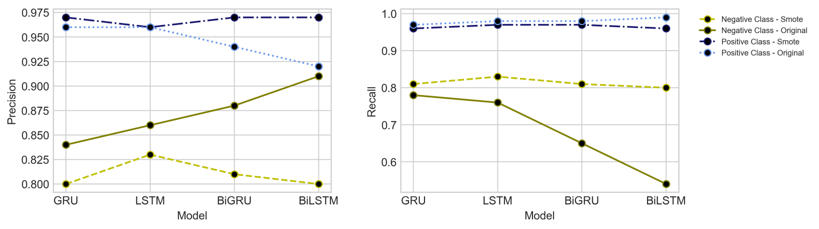

Fig. 1.5 Precision and Recall values of models trained with dataset D2

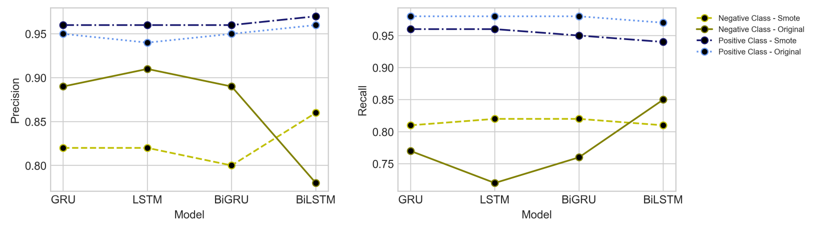

Fig. 1.6 Precision and Recall values of models trained with dataset D3

1.5, it can be observed that there is a drastic decrease in the negative class recall values trained with the original dataset D2, while the SMOTE dataset of D2 provides balanced results of precision and recall values for all the models. This shows that the high values of F1 scores in the case of models trained with the original dataset of D2 are due to the high values of precision and recall scores of only the majority (positive) class.

Similar, the accuracy, F1-score, and AUC of all the four models are slightly high in the SMOTE dataset of D3 than the original dataset. As shown in figure 1.6, the precision and recall values of models trained with original dataset D3 are less for all the models in the case of minority class. These precision and recall values are high and balanced for all the models trained with SMOTE dataset of D3.

The similar values of AUC for every model in their respective dataset column in table 1.2 corresponds to the conclusion that the models trained with both balanced and imbalanced datasets can correctly distinguish the positive and negative class samples. Across the datasets, the lower values of F1-score and AUC for D1 compared to D2 and D3 are because of fewer number data present in D1. Also, as mentioned in section 2.2, the data in D1 is not distinguished, leading to the AUC values around 0.95 while the AUC scores are between 0.96 and 0.97 for D2 and D3.

Among the models and across the datasets, the LSTM models, specifically BiLSTM models, performed comparatively better. For dataset D1, BiLSTM model performs the best among the others with F1 score 89.68% and 0.9585 AUC. BiLSTM model performs the best among the others, even for dataset D2 with an F1 score of 94.61% and 0.9702 AUC. In dataset D3, GRU performs the best with an F1 score of 94.01% and 0.9743 AUC, with BiLSTM being second best with an F1 score of 93.96% and AUC 0.9731. For every model, the SMOTE balanced datasets have given better results than the original imbalanced datasets in all the above cases.

4 Conclusion and Future Work

In this work, deep learning models for binary sentiment classification have been presented with the focus on dealing with the class imbalance of the datasets. The SMOTE method is applied to observe how the data augmentation of text data can affect the performance of the deep learning models. Deep learning models including LSTM, GRU, BiLSTM, and BiGRU are trained with the original imbalanced and SMOTE balanced datasets and presented a comprehensive comparison of the results.

Three datasets with different imbalance ratios are considered to analyze the performance of the models with varying numbers of positive and negative class samples. Unlike in the original datasets, the models trained with the oversampled datasets (SMOTE algorithm) showed a good trade-off between precision and recall scores, with the overall performance comparatively better. The results show that BiLSTMs performed better than other models due to their performance metrics values slightly more significant than that of the other models, across all the three datasets, with balanced and high precision and recall scores for both majority and minority sentiment classes. Thus, it can be concluded that SMOTE increases the models' performance in case of class imbalance.

In future work, the extensions of SMOTE can be experimented. Future work also aim to use pre-trained word embeddings like Glove and Word2Vec, and ensemble deep learning models like Convolution Neural Networks (CNNs) and RNN to find if there can be any significant difference in the models' performances.

References

1. Ah-Pine J, Soriano-Morales E. (2016). A Study of Synthetic Oversampling for Twitter Imbalanced Sentiment Analysis. *DMNLP@PKDD/ECML 1646*: 17-24.
2. Aye YM, Aung SS. (2017). Sentiment analysis for reviews of restaurants in Myanmar text. *18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*: 321-326.
3. Barry J. (2017). Sentiment Analysis of Online Reviews Using Bag-of-Words and LSTM Approaches. *25th Irish Conference on Artificial Intelligence and Cognitive Science*: 272-274.
4. Brownlee J. (2020). *Imbalanced Classification with Python: Better Metrics, Balance Skewed Classes, Cost-Sensitive Learning.* Machine Learning Mastery. https://books.google.be/books?id=jaXJDwAAQBAJ
5. Buitinck L, Louppe G, Blondel M, Pedregosa F, Mueller A, Grisel O, Niculae V, Prettenhofer P, Gramfort A, Grobler J, Layton R, Vanderplas J, Joly A, Holt B, Varoquaux G. (2013). API design for machine learning software: Experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*: 108-122.
6. Chawla N, Bowyer K, Hall L, Kegelmeyer WP (2002) SMOTE: Synthetic Minority Oversampling Technique. *J. Artif. Intell. Res. (JAIR) 16*: 321-357.

7. Cho K, Van Merriënboer B, Bahdanau D, Bengio Y. (2014). On the properties of neural machine translation: encoder-decoder approaches. *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*: 103-111.
8. Chollet F. (2015). Keras.
https://keras.io
9. Feng H, Lin R. (2016). Sentiment classification of food reviews.
https://arxiv.org/abs/1609.01933
10. Fernández A, García S, Galar M, Prati R, Krawczyk B, Herrera F. (2018). *Learning from Imbalanced Data Sets.* Springer.
11. Heikal M, Torki M, El-Makky N. (2018). Sentiment Analysis of Arabic Tweets using Deep Learning. *Procedia Computer Science 142*: 114-122.
12. Hochreiter S, Schmidhuber J. (1997). Long short-term memory. *Neural computation 9*(8): 1735-1780.
13. Johnson J, Khoshgoftaar T. (2019). Survey on deep learning with class imbalance. *Journal of Big Data 6*: 27.
14. Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Prog Artif Intell 5*: 221–232 . doi :10.1007/s13748-016-0094-0
15. Krawczyk B, McInnes BT, Cano A. (2017). Sentiment Classification from Multi-class Imbalanced Twitter Data Using Binarization. *HAIS*: 26–37.
16. Lango M. (2019). Tackling the Problem of Class Imbalance in Multi-class Sentiment Classification: An Experimental Study. *Foundations of Computing and Decision Sciences 44*(2): 151-178.
17. Machine Learning Glossary. (2017).
https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html
18. Mäntylä MV, Graziotin D, Kuutila M. (2018). The Evolution of Sentiment analysis — A Review of Research Topics, Venues, and Top Cited Papers. *Computer Science Review 27*: 16–32.
19. McAuley J, Leskovec J. (2013). From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. *WWW*: 897-908.
20. Ni J, Li J, McAuley J. (2019). Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. *EMNLP*: 188-197.
21. Pang B, Lee L, Vaithyanathan S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*: 79-86.
22. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research 12*: 2825-2830.
23. Prechelt L. (1998). Early Stopping - But When?. *Neural Networks: Tricks of the Trade*: 55-69.
24. Prusa J, Khoshgoftaar TM, Dittman DJ, Napolitano A. (2015). Using Random Undersampling to Alleviate Class Imbalance on Tweet Sentiment Data. *16th IEEE International Conference on Information Reuse and Integration*: 197-202.
25. Tieleman T, Hinton G. (2012). Lecture 6.5 - rmsprop: Divide the gradient by a running average of its recent magnitude.
26. Dablain D., Krawczyk N., Chawla N.V. (2021) "DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data" arXiv:2105.02340v1.
27. Schuster M, Paliwal KK. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing 45*(11): 2673-2681.
28. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research 15*: 1929-1958.
29. Turney P. (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *Computing Research Repository - CORR*: 417-424.
30. Zhang A, Lipton ZC, Li M, Smola AJ. (2020). Dive into Deep Learning.
https://d2l.ai