TEXT AUGMENTATION WITH BERT ON IMBALANCED DATA

# PERFORMANCE EVALUATION OF TEXT AUGMENTATION METHODS WITH BERT ON IMBALANCED DATASETS

---

A Thesis

Presented to

The Faculty of the Graduate School

At the University of Missouri

---

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science in Computer Science

---

by

LINGSHU HU

Dr. Yi Shang, Thesis Supervisor

MAY 2022

TEXT AUGMENTATION WITH BERT ON IMBALANCED DATA

The undersigned, appointed by the dean of the Graduate School, have examined the [thesis] entitled

PERFORMANCE EVALUATION OF TEXT AUGMENTATION METHODS WITH BERT ON IMBALANCED DATASETS

presented by Lingshu Hu,

a candidate for the degree of Master of Science,

and hereby certify that, in their opinion, it is worthy of acceptance.

_____

Professor Yi Shang

_____

Professor Jianlin Cheng

_____

Professor Detelina Marinova

**ACKNOWLEDGEMENTS**

I first would like to thank my advisor, Dr. Yi Shang, and committee member Dr. Jianlin Cheng. They were open-minded and accepted me, a journalism student without a computer science background, into their computer science classes. From them, I became increasingly interested in computer science, especially AI and machine learning, and decided to pursue a degree in the field. I also would like to thank my other committee member, Dr. Detelina Morinova. I worked with her and Dr. Shang's team for one year to study the communication effects of salespeople. During that time, I learned a lot and had the opportunity to connect my computer science skills to my current research focus—business analytics. I have really enjoyed working with this wonderful team and appreciate the help and suggestions I obtained from my fellow team members, such as Can Li and Wenbo Wang. Last but not least, I would like to thank my family and friends. Their company made me feel connected and loved. I would especially like to recognize Yoonjae Shin. He brightened my life and helped me in many ways.

I will always miss my time at the University of Missouri, where I also obtained my Ph.D. in Journalism. I am grateful for the freedom, opportunities, and support the university gave me to explore different interests. It has been one of the most enjoyable times that I have had in my life.

# TABLE OF CONTENTS

# PERFORMANCE EVALUATION OF TEXT AUGMENTATION METHODS WITH BERT ON IMBALANCED DATASETS

Lingshu Hu

Dr. Yi Shang, Thesis Supervisor

## ABSTRACT

Recently deep learning methods have achieved great success in understanding and analyzing text messages. In real-world applications, however, labeled text data are often small-sized and imbalanced in classes due to the high cost of human annotation, limiting the performance of deep learning classifiers. Therefore, this study examines the effectiveness of Word2Vec and WordNet augmentation methods with BERT fine-tuning on datasets of various sizes (e.g., 500, 1,000, and 5,000 training documents) and imbalance ratios (e.g., 4:1 and 9:1). It compares them with other methods for imbalanced data, including boosting, SMOTE, and simple oversampling, combined with widely used machine learning models, including logistic regression, fully connected neural network, and LSTM. Experimental results show that Word2Vec augmentation improves the performance of BERT in detecting the minority class, and the improvement is most significantly (9%-30% recall increase compared to the base model and 11%-12% recall increase compared to the model with the oversampling method) when the data size is small (e.g., 500 training documents) and highly imbalanced (e.g., 9:1). When the data size increases or the imbalance ratio decreases, the improvement generated by the Word2Vec augmentation becomes smaller or insignificant. Moreover, Word2Vec augmentation plus BERT achieves the best performance compared to other models and

methods, demonstrating a promising solution for small-sized, highly imbalanced text classification tasks.

## 1. Introduction

In the internet age, more than millions of text messages are created every day. Individuals, media, organizations, or even robots can generate text information online and on social media. Their voices are recorded and stored as behavior trace-data in the servers of the internet. These data provide researchers great opportunities to study human behaviors in a natural setting, i.e., an environment that is not created or interrupted by researchers. Through analyzing these data, researchers can reveal rich social and psychological meanings behind the texts because the language people use in daily life "reflect what we are paying attention to, what we are thinking about, what we are trying to avoid, how we are feeling, and how we are organizing and analyzing our worlds" [1].

In social science, especially communication studies, analyzing text data has its long tradition. Generally, there are two groups of methods—qualitative and quantitative methods. In qualitative methods such as discourse analysis, researchers read every document in detail and analyze the nuances of language use based on context. In contrast, in quantitative methods such as content analysis, researchers examine hundreds of documents and group them into different categories or mark the presence of some predefined topics. The internet and social media undoubtedly provide researchers rich resources to conduct both qualitative and quantitative text analysis. However, due to the large volume of information generated on these platforms, the traditional methods, which require human lobar to go through the texts, cannot produce representative and generalizable results. Therefore, computational text analysis methods such as machine learning and deep learning have become an increasingly important topic in the field [2].

Text classification is one of the most commonly used computational methods to analyze texts. By assigning texts into different categories, humans can narrow down the scope of the texts, explore the distribution of topics, examine the trends of changes over time, and investigate the interactions between topics and other social or personal variables. Basically, two approaches can be used to classify text: supervised machine learning and unsupervised machine learning. Supervised learning is based on training a statistic model with a dataset that has already been correctly classified [3]. The model learns patterns from the training dataset and can then be used to predict the categories of other unclassified datasets. For example, if we have a dataset with some positive and negative tweets, we can use these labeled data to train a model and then use the model to predict the positivity of a new tweet. In contrast, unsupervised learning does not require pre-labeled training datasets. Algorisms can identify the patterns of data by themselves. For example, if we feed an unsupervised learning algorism some articles from newspapers, it may automatically categorize these articles into political news, sports news, and so forth. However, compared with supervised learning, the usage of unsupervised learning is limited in social science studies because researchers can hardly control what type of patterns they want to detect from the texts. Therefore, supervised learning is preferred if researchers aim to investigate particular patterns of their interest.

### 1.1. Challenges of Supervised Text Classification

Although supervised machine learning has been applied by many social science scholars to categorize and unpack text messages online, two major challenges—small sample size and imbalanced classes—limit its broader use in relevant research.

#### *1.1.1. Small Sample Size*

In terms of supervised learning, first and foremost, pre-labeled text data can hardly be obtained. Scholars usually need to manually read and code texts to generate labeled training datasets, which is time and labor expensive. As a result, it is difficult for researchers to obtain large volumes of annotated text data and feed them to complex machine learning models such as neural networks. Using simpler models could be a solution to small-sized datasets. According to the bias and variance trade-off, simpler models have lower variance and require fewer training data. However, simpler models may have higher bias and not be able to make an accurate classification.

#### *1.1.2. Imbalanced Classes*

Moreover, supervised classification usually requires balanced training data. In other words, training datasets should have an equal number of data points in different categories. Otherwise, the accuracy score may not reflect the real validity of the classification. For instance, if one dataset has 90% negative tweets and 10% positive tweets, the model can predict all tweets as negative and get a 90% accuracy score. In this case, although the accuracy is 90%, the model does not identify any positive tweets and is thus not valid at all.

Because of the challenges described above, developing approaches to boost classifiers' performance on small-sized, imbalanced text datasets is an important topic for researchers. This study will first review the literature in the area, develop a pipeline to combine Word2Vec augmentation and the BERT model, test its performance on datasets of various sample sizes and imbalance ratios, and compare it with other commonly used methods and models. The contribution of this study lies in three aspects: first, it utilizes

transfer learning with text augmentation methods to deal with the imbalance and small size problems; second, it regards sample sizes and imbalance ratios as factors influencing the performance of methods for imbalanced data and examines their effects; and third, it tests the performance of different methods for imbalanced data with different machine learning models and reveals how the model complicity affects the methods for imbalanced data.

## 2. Background and Related Work

Imbalanced data is a common problem in text classification tasks. Previous research suggests that the methods in this area can be grouped into data sampling, algorithm modification, and cost-sensitive learning [4, 5]. However, these three categories might overlap with each other and generate confusion. For instance, cost-sensitive learning can be one type of algorithm modification because it changes the loss function in machine learning models. Therefore, this study simplifies these three categories into sample-based and model-based methods. The former group concentrates on balancing sample distributions or making them more representative [e.g., 4, 6, 7-11]. The second group of methods focuses on developing machine learning classifiers to improve their general performance or performance on imbalanced datasets [e.g., 12, 13, 14].

### 2.1. Sample-Based Methods

Among these two groups of methods, sample-based methods are most commonly used to deal with imbalanced problems. According to the literature, it can be further

divided into three groups—direct-sampling methods, term-weighting methods, and background-knowledge-based methods [4].

#### *2.1.1. Direct-Sampling Methods*

Direct-sampling methods refer to directly changing the data by oversampling or downsampling. The most intuitive approach is to randomly repeat the minority documents or remove some majority documents to force the model to give more weight to the minority class during the training process. However, these methods raise problems such as over-fitting or dropping necessary data [15]. For instance, although deleting data from the majority class may make a classifier perform better at identifying the minority class, it may reduce the classifier's validity because it reduces the representativeness of the sample data. This is particularly true when researchers only have a small-sized dataset. Deleting data points in this case may largely influence the performance of models and limit their prediction power. Accordingly, researchers, especially in the fields where labeled datasets are usually small-sized, typically do not want to remove data (i.e., downsampling) to balance data categories, making over-sampling a preferred method. However, previous research shows that simple repeating data does not significantly improve the performance of minority class recognition [16]. More sophisticated methods are therefore needed to deal with the imbalance problem.

SMOTE, the abbreviation for the "Synthetic Minority Over-sampling Technique," is one of the more sophisticated oversampling approaches. It oversamples minority data to balance classes' ratio in datasets without simply repeating the minority data [7]. Proposed by Chawla and associates [6], this method identifies a minority sample, randomly chooses its k nearest neighbors, calculates the differences between the feature

vector of the sample and its neighbors, multiplies the differences by a random number ranging between 0 and 1, and adds the results back to the sample to form a new data point. For example, consider a dataset with two features. The data point A's feature vector is (5, 2). We find its nearest neighbor B with a feature vector (6, 1). We calculate their differences by using (6, 1) minus (5, 2) and get (1, -1). We multiply (1, -1) with a random number .2 and get (.2, -.2). Then we add (.2, -.2) to sample A (5, 2) and get a new sample (5.2, 1.8). In this way, we create more data points that are similar but different from the minority class data points, which increases our data's representativeness of the minority class. SMOTE has been demonstrated effective in improving minority class recognition [7, 8]. For example, [14] applied SMOTE with logistic regression and support vector machine (SVM) to classify short descriptions of work experiences.

Based on SMOTE, researchers have developed more efficient synthetic methods, such as WEMOTE and CWEMOTE [17]. The basic idea of WEMOTE is that we randomly select two vectors from the minority class and compute their mean vector. This new mean vector is a new instance of the minority class. CWEMOTE takes a further step. It first uses the k-means algorithm to find the clusters within the minority class and then applies WEMOTE on each cluster to generate new instances. The number of new instances of each cluster depends on the weight ($\frac{n_i}{\sum_{j=1}^{m_c} n_j}$, where $n_i$ is the size of cluster i, $m_c$ is the number of clusters) of each cluster.

Besides SMOTE, there are other similar methods to generate synthetic minority samples. For instance, the hidden Markov models (HMM) based oversampling method [18] uses feature words and their weights to synthesize new samples. Boundary region cutting (BRC) algorithm [19] downsample the majority class by randomly eliminating the

samples of the majority class in the dense boundary region. This method can alleviate the boundary ambiguity of two classes and improve classification accuracy.

### *2.1.2. Term-Weighting Methods*

Term-weighting methods give different weights to different feature words, which can balance the importance of feature words. In general, a selected feature set consists of various positive and negative features indicating the membership and non-membership of a class [20]. Adjusting the weights of these features can influence the attention of the classification model toward the classes.

TFIDF is a widely used weighting scheme based on bag-of-words encoding. It is the product of Term Frequency (TF: Number of repetitions of a word in a document / Number of total words in a document) and Inverse Document Frequency (IDF: Log [Number of total documents / Number of documents containing the word]). Compared to bag-of-words or term frequency (TF) encoding, TFIDF is more effective because it gives smaller weights to commonly used words and larger weights to uncommonly used words in the dataset, which better represents the overall condition of the whole datasets [20].

Based on TFIDF, another weighting scheme—Term frequency inverse class frequency (TFICF)—was developed [21]. Instead of calculating inverse document frequency, it calculates the inverse class frequency (ICF), which is the logarithm of the number of classes over the number of classes containing a specific word. In contrast to IDF, ICF operates on the classes of documents instead of individual documents. As such, TFICF can give more weight to those words which better discriminate classes. Similarly, [22] proposed a positive and negative-based term weighting scheme (PNF and PNF2) to consider the class membership.

Though TFIDF and other weighting schemes are relatively simple to understand and implement, they generate sparse high dimensional feature vectors (i.e., many features/variables score zero) and assume independence between each word (i.e., ignoring the order and context information of words). When using a small-sized dataset to train models, sparse high dimensional feature vectors may lead to overfitting.

### *2.1.3. Background-Knowledge-Based Methods*

Background-knowledge-based methods utilize supplementary information to oversample the minority class and improve classification performance on imbalanced text data. For instance, [10] developed "inversion" and "imitation" approaches to create new instances for sentiment classification. Specifically, the "inversion" approach replaces the sentimental words in the majority class with the opposite sentimental words to generate new samples of the minority class. For example, it changes "I like dogs" from the positive sentiment class to "I hate dogs," a negative sentiment case. On the contrary, the "imitation" approach replaces the sentimental words in the minority class with their synonyms to create more instances of that class, e.g., changing "I like dogs" to "I love dogs." However, this method has its limitations. First, it only works with sentiment classification. Moreover, it assumes that the sentiment is solely determined by those sentimental words. If this is the case, we do not need machine learning algorithms to classify sentiment; a sentiment dictionary is enough for the task.

Semantic similarity augmentation [23] is another background-knowledge-based method. Over the recent years, the development of distributed representations (word embeddings) has primarily improved various NLP tasks by better modeling the semantic relations among words in texts [23, 24]. Unlike bag-of-words encoding, which uses one

vector unit (i.e., one variable) to represent a word, the distributed representation uses multiple units to form representations. One of the early and commonly used distributed representation approaches in NLP is Word2Vec [25]. By feeding a large volume of text data into a one-layer neural network to train a model to predict words according to their context words (words around the target word) or vice versa, Word2Vec can generate a vector of weights to represent each word. These vectors can reflect complex semantic relations among words. For example, the word "dog" should be very close to "cat" in the vector space. If we use formula (1.0) to calculate the cosine similarity between the vector of "dog" and the vector of "cat," the similarity score should be relatively higher than that of the vectors of "dog" and "book."

$$S_c(A,B) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1.0)$$

(Where A and B are vectors of two words, and $A_i$ and $B_i$ are components of vectors A and B.)

After obtaining the vectors of words, we can generate new samples of the minority class by replacing some words in the documents of the minority class with words having high semantic similarities with the targe words. This approach is similar to the synonym replacement but can have a more considerable variability because the deep learning model can be trained on a large corpus and include the complex semantic relations among a large volume of words.

Another text augmentation strategy uses knowledge graphs (or semantic networks) to replace synonyms to generate new texts. Knowledge graphs group real-world entities into different categories—such as objects, events, or situations—and illustrate their relationships. WordNet is one of the popular open-source knowledge

graphs for the English language [23]. It categorizes different words such as nouns, verbs, adverbs, and adjectives into different sets of synonyms. Each set expresses similar concepts, provides word definitions and usage examples, and records relations among the words from different synonym sets [23, 26]. ConceptNet is another commonly used knowledge graph that includes around 21 million everyday concepts and their relationships [27, 28]. Scholars have demonstrated the effectiveness of using these knowledge graphs to generate new texts to improve the performance of classification algorithms [e.g., 23, 28, 29].

Round-trip translation (RTT) can also be used to generate new texts. RRT is the process of translating a word, phrase, sentence, or paragraph into another language and then translating them back into the original language [30]. Variations were created through this translating process because the result text of RRT should keep the core meaning but be different from the original text. For example, after translating into Chinese and translating back to English, the sentence “Detective Batman at its peak! Great storyline. Just as dark a universe as we’ve come to expect from DC” becomes “Batman detective pinnacle! Great storyline. A universe as dark as we’ve come to expect from DC.” [31] applied this method to improve the performance of automatic labeling with an imbalanced text dataset. [32] used RTT as a text augmentation method to leverage the target monolingual data in translation. [33] tested the effects of four intermediate languages (French, Spanish, German, and Hindi) with RTT on aggression detection on social media.

Besides semantic similarity augmentation, synonym replacement, and round-trip translation, there are still other text augmentation or generation methods, such as random

deletion—randomly removing words in documents with pre-defined probability, random insertion—randomly adding words in documents with pre-defined probability, and random switching—randomly switch the orders of some words in documents [34]. The benefits of text augmentation methods are that they generate new texts that can be used with any classification model. They can also increase the sample size of balanced data, which solves the small sample size problems. However, some scholars argue that text augmentation methods may not generate significant improvements when using pre-trained models or contextual word embeddings [12, 34]. This study therefore aims to test if text augmentation methods can generate better results with the pre-trained BERT model on imbalanced, small-sized datasets.

## 2.2. Model-Based Methods

Different machine learning models have different performances. To deal with imbalanced, small-sized data, scholars also develop various algorithms to improve the performance of machine learning models.

### *2.2.1. Cost-Sensitive Approaches*

Boosting is one of the most commonly used ensemble methods in machine learning. Ensembling refers to a group of techniques that constructs multiple models and takes the combination of their outputs as the overall output. Ensemble methods can improve models' accuracy and reduce the prediction error rate by reducing variance, bias, or both [35].

As one of the popular ensemble methods, boosting combines multiple "weak" classifiers to form a committee that outperforms any of the weak classifiers [36]. The rationale for boosting is simple: it iteratively trains the model and gives larger weights to

misclassified data points. Specifically, the algorism first assigns each data point (e.g., each tweet in a Twitter dataset) an initial weight. Then it runs the base classifier, such as Naïve Bayes, to the data and gets an error rate. Based on the previously trained classifier's performance, the algorism adjusts the weight of each data point by increasing the weight of the misclassified data points. Then it trains a new classifier on the data again and repeats the above process. Finally, after enough base classifiers have been trained, the algorism combines them together to "form a committee using coefficients that give different weight to different base classifiers" [36].

$$\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(X_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}} \quad (2.0)$$

In formula 2.0, $I$ (…) function serves as a switch. When the predicted value $y_m$ given by the data point $X_n$ does not equal the true value $t_n$, it generates 1; when $y_m$ equals $t_n$, it generates 0. Then we time the value generated by the $I$ (…) function to the pre-assigned weight $w_n$. We add up all weights and normalize it by dividing the total weights. We use this result to further update the data weighting coefficients.

By combining "weak" classifiers, boosting can add non-linear functions in the model, which increases the model complicity and might increase the prediction power. In addition, by increasing the weights of misclassified data points, boosting forces the model to focus more on misclassified minority data, which is supposed to improve the model's performance on imbalanced datasets.

Boosting has been demonstrated to be successful in improving different classifiers' performance by many scholars [e.g., 37, 38]. Also, scholars have applied boosting algorism to different tasks, including text classification. For instance, Kudo and Matsumoto [39] used boosting classification to classify semi-structured text in Japanese;

Ehrentraut and colleagues [40] used boosting algorism to detect documents revealing hospital-acquired infections; Sun and colleagues [13] used boosting algorism to solve the problem of imbalanced data in text classification. Therefore, this study examines whether boosting can be used to improve the performance of text classification of various sample sizes and imbalance ratios.

Besides boosting, researchers can also modify the classification threshold to encourage the classifiers to focus more on the minority class. Classifiers return probabilities associated with labels when predicting classes. These probabilities represent the confidence of the prediction. The threshold for binary classification is usually 0.5. When the probability is over 0.5, the model predicts one class; when it is below 0.5, it predicts the other. This threshold may not be optimal when dealing with imbalanced data since the minority class is more likely to be ignored by the model. Accordingly, adjusting the threshold to favor the minority class can be an approach to make classifiers cost-sensitive and improve the classification performance [41].

Similarly, researchers can also modify the loss function to make the classifiers cost-sensitive and improve their performance on imbalanced data. Specifically, researchers can give weight to the loss function to make it cost-sensitive to the minority class. For example, the original sigmoid loss function $loss(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right)$ can be modified to $loss(x, class) = -weight[class]\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right)$. This modification gives classifiers opportunities to penalize the misclassification of small classes, which are usually misidentified as majority classes according to the maximum likelihood. [12] applied this method to the

BERT model. They changed the loss function of the last fully connected layer and found it improved the performance of BERT fine-tuning.

#### *2.2.2. Deep Learning and Transfer Learning*

Generally, a more advanced and complex model can detect more detailed patterns and may have better classification performance. Over the last ten years, neural networks and deep learning have outperformed many classifiers such as support vector machines in various classification tasks. A neural network can be regarded as stacks of many simple classifiers. By combining them hierarchically, the neural network can detect some features from simple to complex. However, according to the bias and variance trade-off, complex models have higher variance and require more training data. As aforementioned, human annotation is expensive and time-consuming. It is possible that some researchers only obtain several hundred of training data. In this case, complex models may easily overfit the training data and produce poor results for the testing data.

Transfer learning, which refers to the method that transfers the already learned knowledge from a related task to a new task [42], can solve this problem. Since transfer learning models are pre-trained on another dataset, it can utilize the supplementary information outside of the training data to improve the model performance. For example, [43] apply transfer learning to solve the imbalance problem by first training a model with balanced data and then to fine-tune the model with imbalanced data. However, this method requires researchers to have balanced data first.

Word2Vec [25], which is mentioned before, is one typical transfer learning model. It can be trained with a large corpus such as Wikipedia or Google news, which contains billions of words, and learns the semantic relations among the words from this

corpus. Then, the learned semantic representations of words can be used in a classification model with several hundreds of documents. Word2Vec has been demonstrated to be effective and achieved state-of-the-art performance on many NLP tasks [25].

Compared to Word2Vec, BERT (Bidirectional Encoder Representations from Transformers) [44] has a more complicated neural network structure and outperforms Word2Vec in various NLP tasks [44]. The BERT model was trained on BooksCorpus (800M words) and English Wikipedia (2,500M words) [44]. It is one type of transformer model, which means besides taking the order of words into account, BERT applies an attention mechanism to avoid information loss from long chains. Discussing the technical details of BERT is beyond the scope of this study. Instead, this study examines the effectiveness of fine-tuning the pre-trained BERT model with methods for imbalanced data on small-sized, imbalanced datasets.

## 3. Methods

This study used Python language and Python APIs such as Scikit-learn [45], Keras [46], and Transformers [47] to test the performance of different approaches on imbalanced, small-sized text classification tasks.

### 3.1. Classification Models

Four models were tested on two datasets of various sample sizes and imbalance ratios. From simple to complicated models, this study applied logistic regression, fully connected neural network, LSTM, and BERT.

### *3.1.1. Logistic Regression*

Logistic regression was selected in this study because it is simple and straightforward. Based on the bias and variance trade-off, simpler models have lower variance and require fewer training data. Considering small-sized datasets used in this study, logistic regression may perform better than more complex deep learning models. Moreover, as a linear model, logistic regression can clearly show how each feature influences the classification results. This characteristic is important to social science studies because the primary concern of social science is the relationships among different factors. Accordingly, due to the simplicity and high explainability of logistic regression, it is one of the most commonly used classifiers in social science studies and has been familiar to social science scholars.

### *3.1.2. Fully Connected Neural Network (FC-NN)*

Although logistic regression has advantages, it cannot handle complex non-linear relationships. Therefore, neural networks may outperform logistic regression in complex classification tasks. Basically, fully connected neural networks can be regarded as stacks of many logistic regressions. Each logistic regression is one neutron to make a binary decision. Combining multiple neutrons, especially combining them in a hierarchical manner, more complicated decisions can be made by the model.

The structure of the neural network is shown in figure 5.

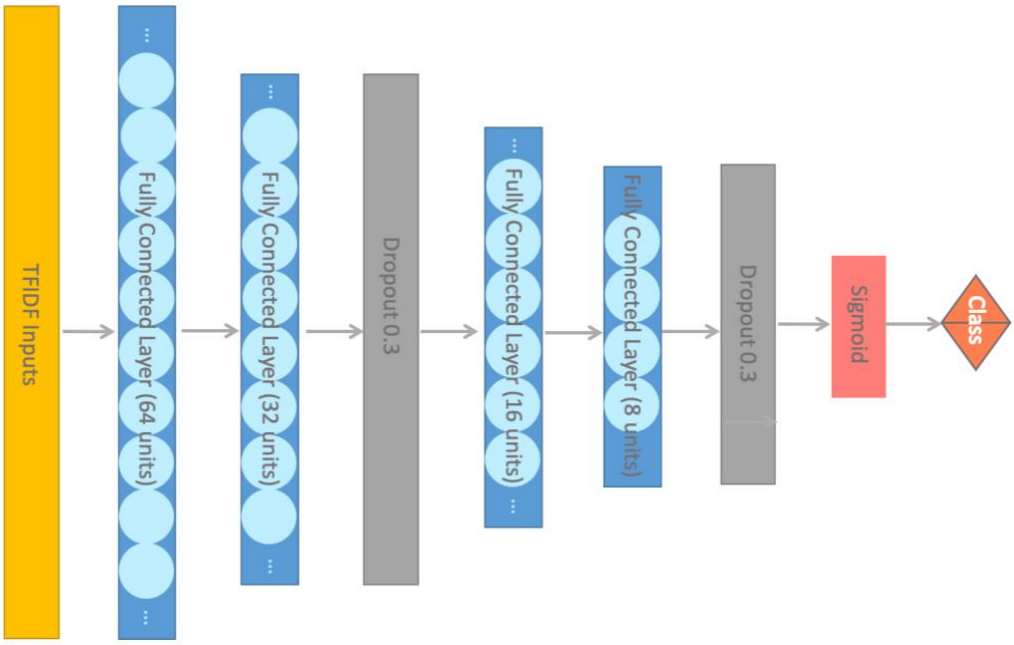


*Figure 1 The Architecture of the Fully Connected Neural Network*

The input for the fully connected neural network is the same as that for the logistic regression, which is 768-dimension TFIDF vectors. The batch size was 128, and the model was trained in 16 epochs. During the training process, the validation accuracy was calculated by using the validation dataset after each epoch, and then the model with the best validation accuracy was selected as the final model.

### *3.1.3. Long Short Term Memory (LSTM)*

Long Short Term Memory (LSTM) [48] is one of the most commonly used recurrent neural networks. Compared to fully connected neural networks, LSTM can take the order information into account. In natural languages, different word orders can have different meanings. Including the order information in the model reflects the nature of the language and can provide more context. Therefore, LSTM can be a better model than fully connected neural networks in text classification.

The structure of the LSTM model is shown in figure 6.

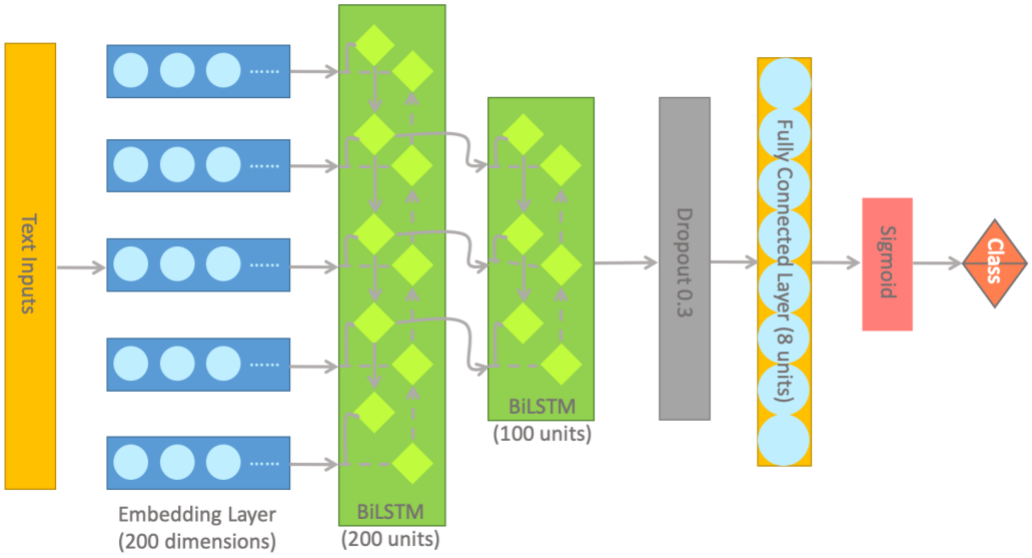


*Figure 2 The Architecture of LSTM*

The input for LSTM is word tokens. Then each token was embedded as a 200-dimension vector in the Keras embedding layer. The vocabulary size was set as 20,000. The batch size was 128, and the model was trained in 16 epochs. During the training process, the validation accuracy was calculated by using the validation dataset after each epoch, and then the model with the best validation accuracy was selected as the final model.

### *3.1.4. BERT*

This study fine-tuned the pre-trained uncased base BERT model [44] by using the Python library “Transformers” [47] with Keras [46]. The batch size was 128, and the model was trained in 16 epochs. During the training process, the validation accuracy was

calculated by using the validation dataset after each epoch, and then the model with the best validation accuracy was selected as the final model.

### 3.2. Machine Learning Methods for Imbalanced Training Data

Five methods for imbalanced training data were tested with the above-mentioned machine learning models. Because different models require different input formats, not every model can work with these methods. Specifically, boosting was only applied to the logistic regression; SMOTE was applied to both logistic regression and fully connected neural network; simple oversampling, Word2Vec, and WordNet augmentations were applied to all models.

#### *3.2.1. Boosting*

Boosting can adjust the weights of misclassified instances and benefit the simple classifiers. Specifically, this study used the AdaBoost algorithm from the Scikit-learn Python API [45], which has been proven to work well on binary classification problems [37]. AdaBoost, short for "adaptive boosting," was developed by Freund and Schapire [49] and is one of the most widely used boosting algorism. Since boosting was developed to improve the performance of simple classifiers, this study only applied it with logistic regression. 4,000 iterations were run to test its effectiveness.

#### *3.2.2. SMOTE*

As aforementioned, SMOTE synthesizes new minority instances by generating vectors from randomly selected instances and their nearest neighbors. SMOTE requires each document to be presented by a one-dimensional vector and produces new one-dimensional vectors for synthetic instances. This requirement makes SMOTE unsuitable to BERT and LSTM models, which require word tokens or word level vectors as input.

Therefore, this study only tested SMOTE with logistic regression and fully connected neural network models. Specifically, this study applied SMOTE methods by using the "imblearn" Python library [50]. Text data were first transformed into TFIDF vectors, It selects five nearest neighbors of every document in the minority class, and generates the new instances based on these five nearest neighbors.

### *3.2.3. Word2Vec Text Augmentation*

Word2Vec text augmentation generates texts data directly and thus can be applied to any models regardless of what input format the models require. The Word2Vec used in this study was pre-trained on Google News Corpus, which has 3 billion running words. The pre-trained Word2Vec model contains 3 million 300-dimension English word vectors [51]. Specifically, this study used "textaugment" Python API [23] to implement the Word2Vec augmentation. All documents in the minority class were selected as sources for augmentation. 90% of random words in every document were selected for replacement. If a selected word is in the Word2Vec corpus, then one random similar word is chosen from the ten most similar words in the corpus to replace this word. If the imbalance ratio is 4:1, then the Word2Vec augmentation is repeated three times to generate three documents for every minority document. If the imbalance ratio is 9:1, the Word2Vec augmentation is repeated eight times to generate eight documents for every minority document. The generated documents were combined with the original documents to make the training dataset balanced.

### *3.2.4. WordNet Text Augmentation*

WordNet text augmentation is similar to Word2Vec augmentation. It can also be applied to every model. The "textaugment" Python API [23] was used again to

implement the method. It first labels words in documents with pos-tags, and then replaces 90% of nouns and verbs with random synonyms in WordNet. Similar to Word2Vec augmentation, WordNet augmentation was run multiple times to generate the minority documents to make the training data balanced.

Table 1 shows three sentences generated by Word2Vec and WordNet. From it, we can see that the replacements made by WordNet are fewer than Word2Vec due to the limited vocabularies of WordNet. Compared to WordNet, the sentences generated by Word2Vec have much more variances, which may benefit complex models.

| Original | Word2Vec | WordNet |
|---|---|---|
| Adrian Pasdar is excellent! Is this film he makes fascinating woman? | jensen pasdar makes superb appears that cinema his made endlessly fascinating teenage girl | adrian pasdar embody excellent embody this picture he cause beguile woman |
| | nigel pasdar was great isn't a another movie i is facinating policewoman | adrian pasdar follow excellent embody this pic he cause beguile charwoman |
| | joel pasdar remains excellent remains every vasanthabalan he made revelatory lady | adrian pasdar embody excellent embody this pic he cause beguile charwoman |

*Table 1 Example Sentences Generated by Word2Vec and WordNet Augmentation*

Word2Vec augmentation method is more time costly than WordNet. It took around 29 hours to generate three times new texts from 1,000 documents from the IMDB dataset and 35 hours to generate eight times new texts from 5,00 documents from the IMDB dataset. Since the Toxic Comments dataset has shorter documents than the IMDB dataset, the Word2Vec augmentation method took 5 hours to generate three times new texts from 1,000 documents and eight times new texts from 5,00 documents from the

Toxic Comments dataset. WordNet only took around 20 seconds to generate these new instances.

#### *3.2.5. Simple Oversampling*

The simple oversampling method also directly works on texts and can be applied to any model. When the imbalance ratio is 4:1, the minority documents in the training dataset were repeated three times to match the number of the majority documents. When the imbalance ratio is 9:1, the minority documents were repeated eight times. The validation and testing datasets were kept original.

### 3.3. Experimental Pipelines

This study combined the above-mentioned models and methods and tested their performance on two datasets with different sample sizes and imbalance ratios. The combination of models and imbalance improving methods are shown in table 2.

| | Logistic Regression | FC-NN | LSTM | BERT |
|---|---|---|---|---|
| Baseline | Yes | Yes | Yes | Yes |
| Oversample | Yes | Yes | Yes | Yes |
| Word2Vec | Yes | Yes | Yes | Yes |
| WordNet | Yes | Yes | Yes | Yes |
| SMOTE | Yes | Yes | No | No |
| Boosting | Yes | No | No | No |

*Table 2 The Combinations of Models and Methods for Imbalanced Data*

Text data were pre-processed by removing punctuations and convert into lowercase. For LSTM and BERT models, text data were tokenized as input. For logistic regression and FC-NN, stop words were further removed from texts, and all words were reduced to their stems. Then 768 most frequently used words in training data were

extracted as features to form TFIDF vectors representing documents. The TFIDF vectors were fed into the logistic regression and FC-NN as input.

This study has three pipelines, as shown in figure 3-5. The first pipeline (figure 3) used text augmentation or simple oversampling methods to increase the number of minority documents and make the training data balanced. For LSTM and BERT, the balanced training text data were tokenized and fed into the models directly as input. For logistic regression and FC-NN, the training text data were further transferred into TFIDF vectors and then fed into models as input.

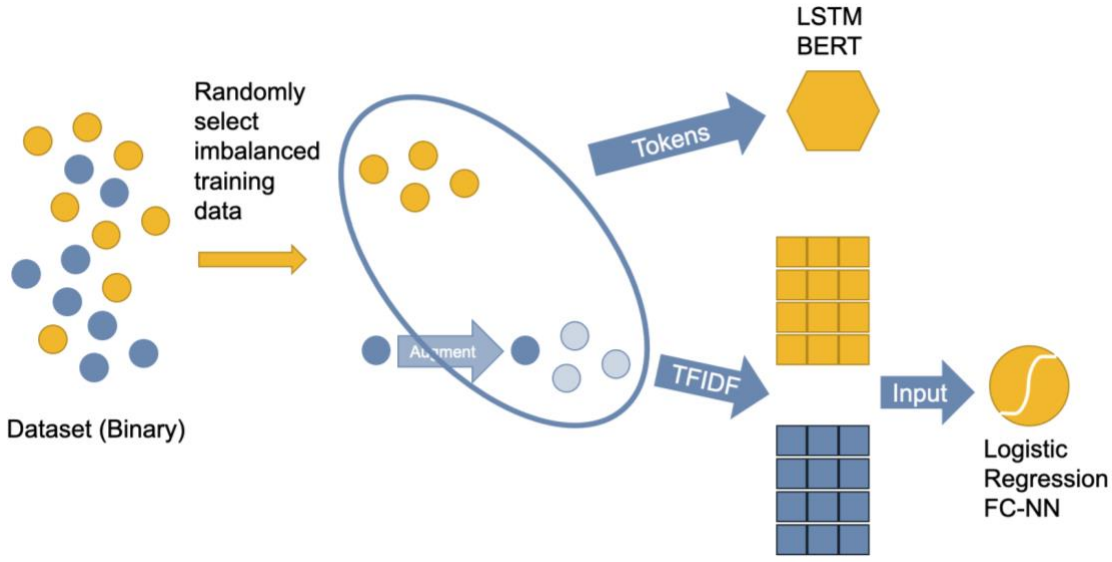


*Figure 3 The Pipeline of Text Augmentation and Simple Oversampling with Logistic Regression, FC-NN, LSTM, and BERT*

The second pipeline (figure 4) shows how SMOTE was implemented. First, random imbalanced training text data were selected and transformed into TFIDF vectors. SMOTE was applied to these TFIDF vectors. It increased the number of minority TFIDF vectors to balance the two classes in the training data. Then these vectors were fed into

logistic regressions and FC-NN models.

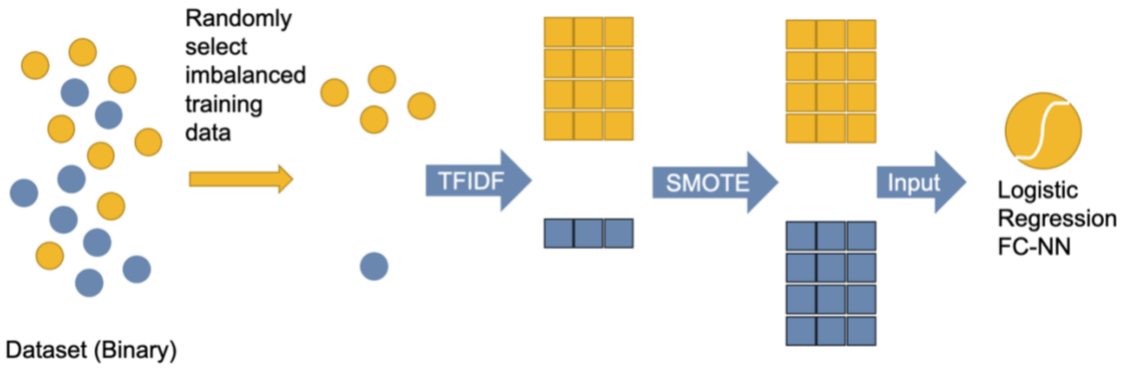


*Figure 4 The Pipeline of SMOTE with Logistic Regression and FC-NN*

The third pipeline (figure 5) shows how boosting works in the experiment. Again, random imbalanced text data were selected and transformed into TFIDF vectors. Then these imbalanced vector data were fed into the boosting algorithm with logistic regression as its classifier.

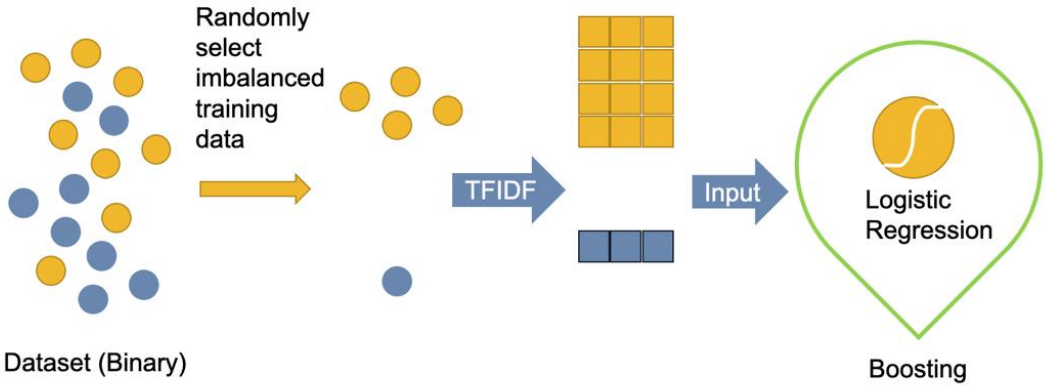


*Figure 5 The Pipeline of Boosting with Logistic Regression*

### 3.4. Evaluation Metrics

Accuracy is the most commonly used metric for evaluating classification performance. However, in an imbalanced dataset, accuracy may be biased and not accurately reflect the performance of models [52]. Therefore, aligning with previous

studies [4, 12, 52], this research used F1 score (formula 3.0) and minority class recall (formula 4.0) as metrics. Among these metrics, minority class recall is of the most interest because it shows the model's ability to identify the minority class regardless of whether the testing data is imbalanced.

$$F1 = \frac{TP}{TP+\frac{1}{2}(FP+FN)} \tag{3.0}$$

$$Recall = \frac{TP}{TP+FN} \tag{4.0}$$

## 4. Results and Discussion

### 4.1. Datasets

IMDB Reviews [53] and Toxic Comments [54] were selected as data to test different models' performance. IMDB Reviews is a wildly used open-access dataset. Researchers can easily replicate this study's experiments on this dataset. IMDB reviews dataset contains two classes—positive and negative reviews. The training dataset has 12,500 positive reviews and 12,500 negative reviews, and the testing dataset also has 12,500 positive reviews and 12,500 negative reviews.

The average length of the IMDB reviews is 231.15, and the median length is 173. The distribution of the length is shown in figure 1. Since the RAM capacity of the computer used in this study is 32 G, which only allows the BERT model to deal with around 170 tokens, this study filtered out reviews of over 170 tokens after text preprocessing, leaving 6,283 positive and 6,271 negative reviews in the training dataset, and 6,489 positive and 6,308 negative reviews in the testing dataset. The length distribution after filtering out is shown in figure 2.

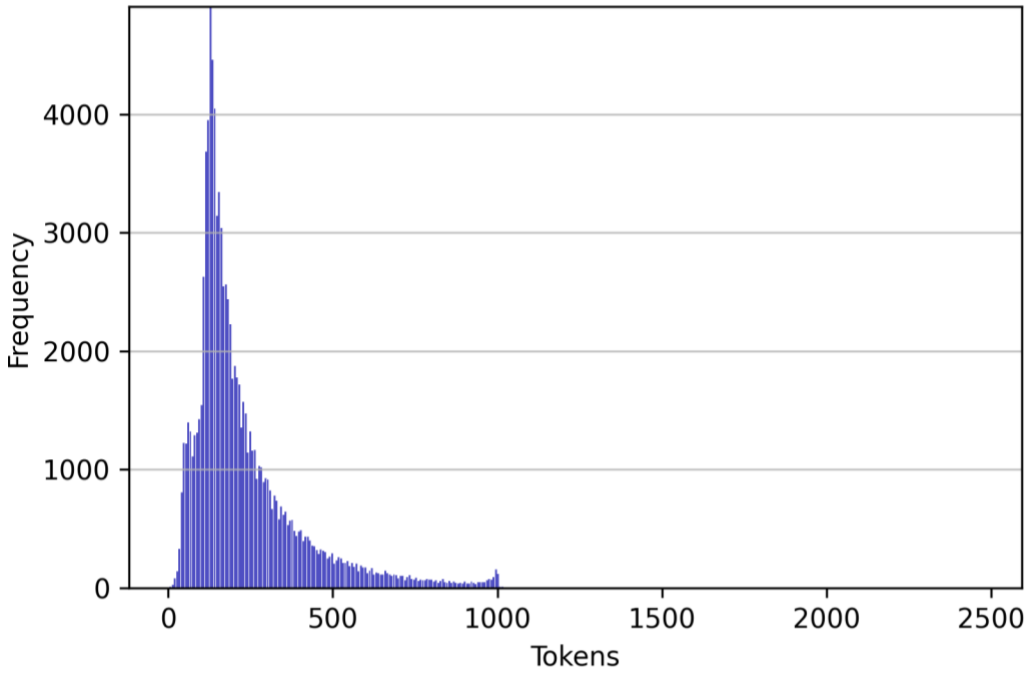


*Figure 6 The Distribution of Document Length in IMDB Reviews Dataset*

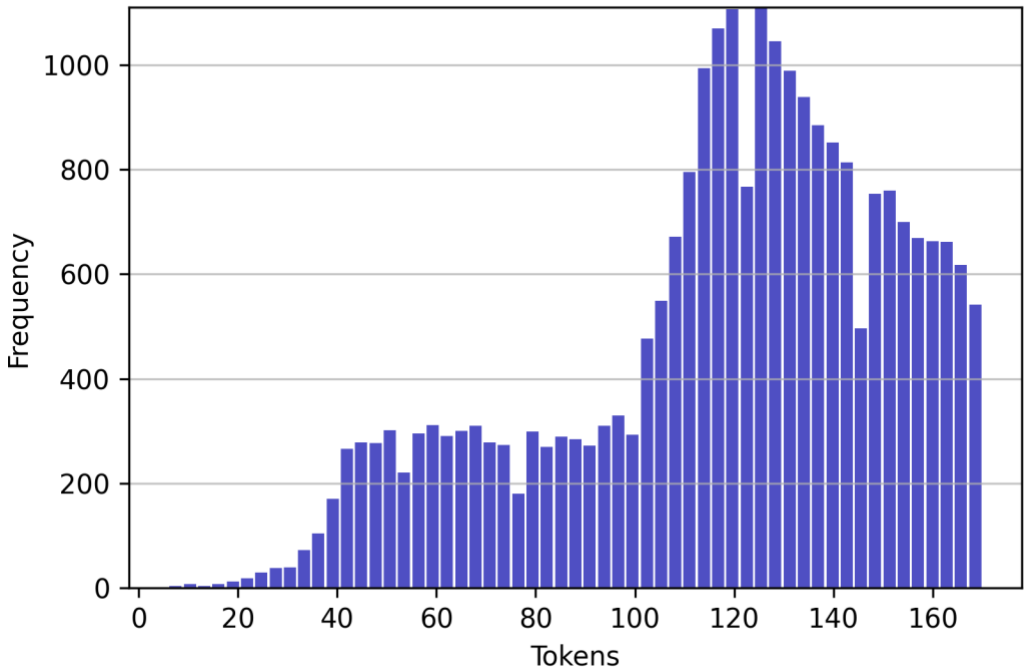

*Figure 7 The Distribution of Document Length in IMDB Reviews Dataset after Filtering out Long Documents*

Since this study aims to examine the performance of classifying imbalanced, small-sized datasets, the imbalanced number of training data of various sizes were manually created as follows:

| Ratio | Size | Majority Class | Minority Class |
|---|---|---|---|
| 4:1 | 500 | 400 | 100 |
| | 1000 | 800 | 200 |
| | 5000 | 4000 | 1000 |
| 9:1 | 500 | 450 | 50 |
| | 1000 | 900 | 100 |
| | 5000 | 4500 | 500 |

*Table 3 Training Sample Sizes and Imbalance Ratios*

The testing dataset was kept the same, with 6,489 positive and 6,308 negative reviews.

This study selected 500, 1,000, and 5,000 sample sizes because they are the numbers that are feasible for human annotation by only a few coders. In the literature of communication and social science studies, scholars usually code several hundreds of documents to analyze their content. There are a large number of content analysis studies in the communication field, which can potentially provide rich resources for machine learning.

Toxic Comments [54] is a dataset used for a public competition on Kaggle. It provides “a large number of Wikipedia comments which have been labeled by human raters for toxic behavior” [54]. The dataset has six labels for comments—non-toxic, toxic, severe toxic, obscene, threat, insult, and identity hate. To simplify the classification task, this study grouped all the comments into toxic and non-toxic comments, making it a binary classification task. This grouping method has been used by other studies such as [52]. After grouping, the training dataset contains 159,571 documents with 89.83% non-toxic and 10.17% toxic comments; the testing dataset contains 63,978 documents with 90.24% non-toxic and 9.76% toxic comments.

The average length of the toxic comments is 66.59, and the median length is 35. The distribution of length is shown in figure 3. After filtering out comments over 170 tokens, the training dataset has 131,525 non-toxic and 15,406 toxic comments, and the testing dataset has 53,209 non-toxic and 6,028 toxic comments. The length distribution after filtering out is shown in figure 4. Then random comments were selected from the training dataset to form new imbalanced training data according to the sizes and ratios in

table 3.

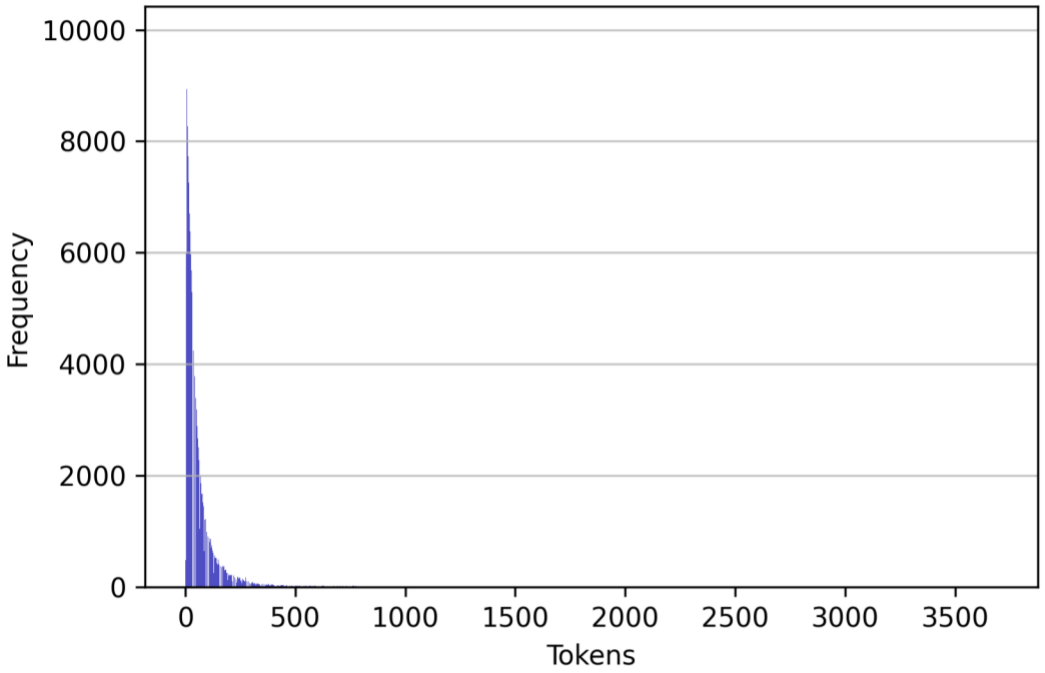


*Figure 8 The Distribution of Document Length in Toxic Comments Dataset*

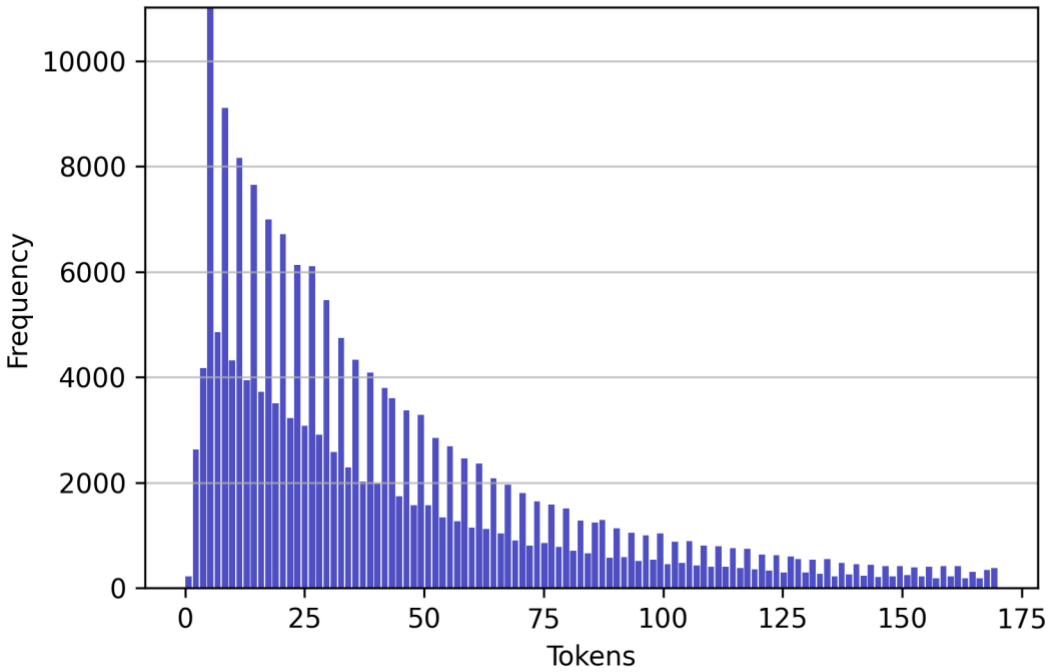

*Figure 9 The Distribution of Document Length in Toxic Comments Dataset after Filtering out Long Documents*

### 4.2 Training Time

The training time of FC-NN, LSTM, and BERT models are shown below:

| | FC-NN | LSTM | BERT |
|---|---|---|---|
| Device | CPU: Intel I3 8100<br>GPU: GeForce 1070<br>RAM: 16 G | CPU: Intel I3 8100<br>GPU: GeForce 1070<br>RAM: 16 G | CPU: Intel Xeon<br>GPU: Google TPU<br>RAM: 32 G |
| Sample Size | | | |
| 500 | 1.4 seconds | 40 seconds | 45 minutes |
| 1,000 | 1.8 seconds | 1.2 minutes | 92 minutes |
| 5,000 | 4 seconds | 6 minutes | 493 minutes |

*Table 4 The Training Time of Fully Connected Neural Network (FC-NN), LSTM, and BERT Models*

After training, the models were tested on the same testing dataset. The IMDB testing dataset is balanced, while the Toxic Comments testing dataset is imbalanced (89.8% majority class and 10.2% minority class).

### 4.3. Experimental Results

#### *4.3.1. IMDB Reviews Data Results*

First, logistic regression was applied with different methods for imbalanced data. The results (see table 5 and figure 10) show that the base logistic regression could not recognize the minority class when the data size is small—the recall and F1 scores are almost zero. When the sample size increases to 5,000, the base logistic regression achieves a 57.9% F1 score and 41.2% recall for the 4:1 imbalance ratio data, while a

20.9% F1 score and 11.7% recall for the 9:1 imbalance ratio data, indicating that this model is sensitive to the imbalance ratio.

The oversampling method generates the best performance combined with logistic regression, followed by SMOTE. The performance improvement is most significant (the F1 score and recall increased by 62% compared to the base model) when the sample size is 5,000 and the imbalance ratio is 9:1. Boosting, Word2Vec, and WordNet augmentations also generate a positive effect, though the performance increase is less significant than the oversampling method.

| Imbalance Ratio | Data size | Methods | Accuracy | F1 | Precision | Recall |
|---|---|---|---|---|---|---|
| 4:1 | 500 | Base | 0.496 | 0.014 | 0.979 | 0.007 |
| | | Oversample | 0.740 | 0.699 | 0.844 | 0.596 |
| | | Word2Vec | 0.582 | 0.336 | 0.863 | 0.209 |
| | | WordNet | 0.539 | 0.177 | 0.945 | 0.097 |
| | | SMOTE | 0.718 | 0.653 | 0.867 | 0.523 |
| | | Boosting | 0.612 | 0.395 | 0.941 | 0.250 |
| | 1000 | Base | 0.519 | 0.101 | 0.977 | 0.053 |
| | | Oversample | 0.782 | 0.760 | 0.863 | 0.679 |
| | | Word2Vec | 0.594 | 0.353 | 0.922 | 0.218 |
| | | WordNet | 0.558 | 0.235 | 0.960 | 0.134 |
| | | SMOTE | 0.764 | 0.731 | 0.866 | 0.633 |
| | | Boosting | 0.665 | 0.523 | 0.945 | 0.361 |
| | 5000 | Base | 0.696 | 0.579 | 0.971 | 0.412 |
| | | Oversample | 0.839 | 0.837 | 0.863 | 0.812 |
| | | Word2Vec | 0.713 | 0.621 | 0.941 | 0.463 |
| | | WordNet | 0.712 | 0.614 | 0.961 | 0.451 |
| | | SMOTE | 0.828 | 0.819 | 0.877 | 0.769 |
| | | Boosting | 0.755 | 0.694 | 0.946 | 0.549 |
| 9:1 | 500 | Base | 0.493 | 0.000 | - | 0.000 |
| | | Oversample | 0.610 | 0.407 | 0.887 | 0.264 |

| | | | | | |
|---|---|---|---|---|---|
| | Word2Vec | 0.507 | 0.070 | 0.803 | 0.036 |
| | WordNet | 0.499 | 0.023 | 0.950 | 0.012 |
| | SMOTE | 0.611 | 0.397 | 0.931 | 0.252 |
| | Boosting | 0.500 | 0.029 | 1.000 | 0.015 |
| 1000 | Base | 0.493 | 0.002 | 1.000 | 0.001 |
| | Oversample | 0.687 | 0.586 | 0.887 | 0.438 |
| | Word2Vec | 0.522 | 0.123 | 0.872 | 0.066 |
| | WordNet | 0.506 | 0.053 | 0.946 | 0.027 |
| | SMOTE | 0.681 | 0.569 | 0.905 | 0.415 |
| | Boosting | 0.554 | 0.219 | 0.974 | 0.124 |
| 5000 | Base | 0.552 | 0.209 | 0.988 | 0.117 |
| | Oversample | 0.816 | 0.802 | 0.880 | 0.737 |
| | Word2Vec | 0.581 | 0.307 | 0.956 | 0.183 |
| | WordNet | 0.577 | 0.290 | 0.977 | 0.170 |
| | SMOTE | 0.808 | 0.789 | 0.893 | 0.706 |
| | Boosting | 0.661 | 0.508 | 0.964 | 0.345 |

*Table 5 Results of Logistic Regression with Different Methods for Imbalanced Data Trained with the IMDB Reviews Dataset (the highlight indicates the best recall and F1*

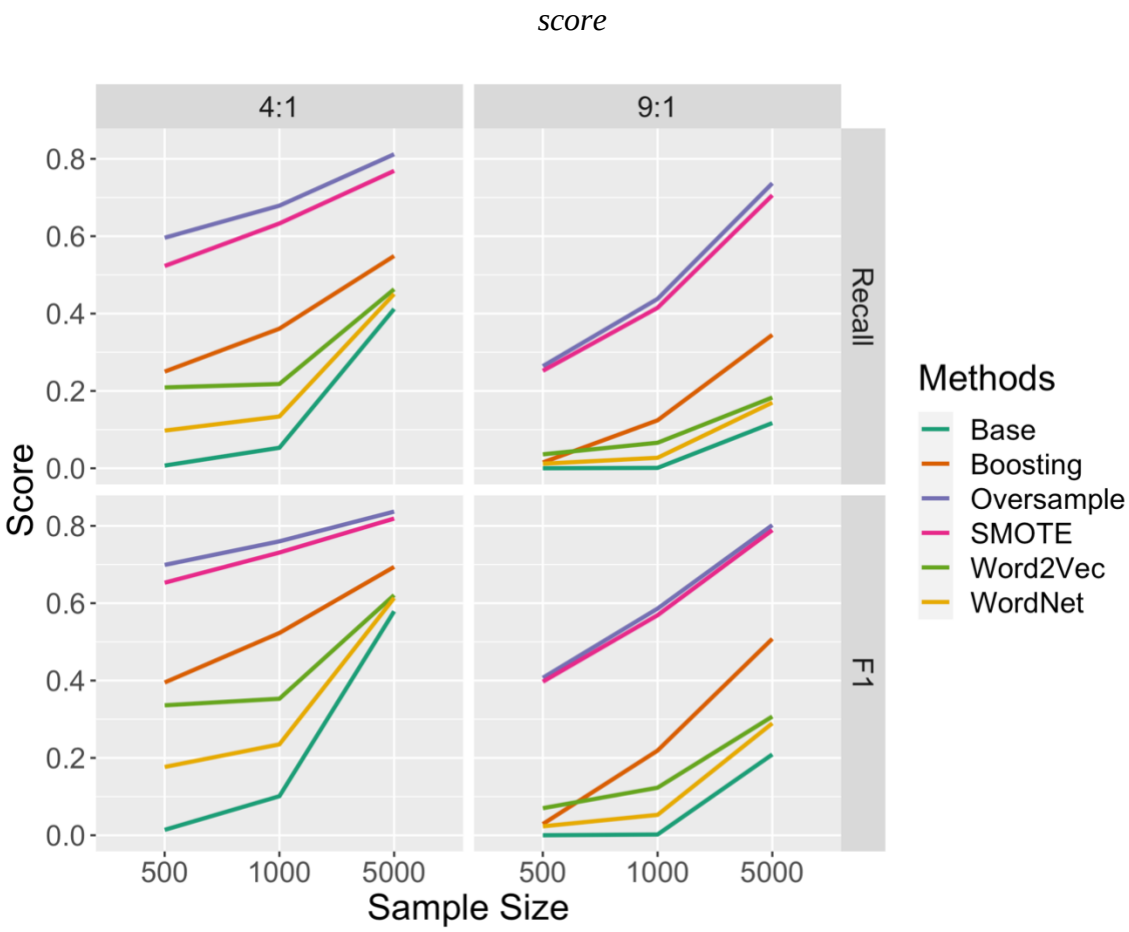


*Figure 10 Recalls and F1 Scores of Logistic Regression with Different Methods for Imbalanced Data Trained with the IMDB Reviews Dataset*

The base model of FC-NN shows a similar pattern (see table 6 and figure 11). It does not work with the 9:1 ratio data even when the data size increases to 5,000. The oversampling method generally achieves the best performance in terms of recall and F1 score when the imbalance ratio is 4:1, while SMOTE performs the best when the imbalance ratio is 9:1.

Compared to the performance in the logistic regression, Word2Vec and WordNet augmentations generate better results in the fully connected neural network. Particularly,

Word2Vec's performance is almost equal to that of the oversampling and SMOTE methods.

| Imbalance Ratio | Data size | Methods | Accuracy | F1 | Precision | Recall |
|---|---|---|---|---|---|---|
| 4:1 | 500 | Base | 0.493 | 0.000 | - | 0.000 |
| | | Oversample | 0.720 | 0.660 | 0.859 | 0.536 |
| | | Word2Vec | 0.711 | 0.647 | 0.849 | 0.523 |
| | | WordNet | 0.574 | 0.298 | 0.904 | 0.178 |
| | | SMOTE | 0.688 | 0.591 | 0.882 | 0.444 |
| | 1000 | Base | 0.493 | 0.000 | 0.000 | 0.000 |
| | | Oversample | 0.723 | 0.652 | 0.899 | 0.512 |
| | | Word2Vec | 0.723 | 0.660 | 0.874 | 0.530 |
| | | WordNet | 0.590 | 0.339 | 0.926 | 0.207 |
| | | SMOTE | 0.779 | 0.798 | 0.743 | 0.863 |
| | 5000 | Base | 0.799 | 0.769 | 0.919 | 0.662 |
| | | Oversample | 0.830 | 0.820 | 0.885 | 0.763 |
| | | Word2Vec | 0.772 | 0.729 | 0.919 | 0.604 |
| | | WordNet | 0.607 | 0.388 | 0.917 | 0.246 |
| | | SMOTE | 0.821 | 0.805 | 0.903 | 0.725 |
| 9:1 | 500 | Base | 0.493 | 0.000 | - | 0.000 |
| | | Oversample | 0.539 | 0.179 | 0.920 | 0.099 |
| | | Word2Vec | 0.533 | 0.163 | 0.885 | 0.090 |
| | | WordNet | 0.521 | 0.114 | 0.920 | 0.061 |
| | | SMOTE | 0.555 | 0.230 | 0.942 | 0.131 |
| | 1000 | Base | 0.493 | 0.000 | - | 0.000 |
| | | Oversample | 0.597 | 0.361 | 0.920 | 0.224 |
| | | Word2Vec | 0.600 | 0.387 | 0.864 | 0.250 |
| | | WordNet | 0.549 | 0.216 | 0.914 | 0.122 |
| | | SMOTE | 0.617 | 0.418 | 0.914 | 0.271 |
| | 5000 | Base | 0.493 | 0.000 | 0.000 | 0.000 |
| | | Oversample | 0.792 | 0.765 | 0.899 | 0.665 |
| | | Word2Vec | 0.616 | 0.408 | 0.936 | 0.261 |
| | | WordNet | 0.603 | 0.378 | 0.918 | 0.238 |

| | | | | | |
|---|---|---|---|---|---|
| | SMOTE | 0.795 | 0.768 | 0.902 | 0.668 |

*Table 6 Results of the Fully Connected Neural Network with Different Methods for Imbalanced Data Trained with the IMDB Reviews Dataset (the highlight indicates the best recall and F1 score)*

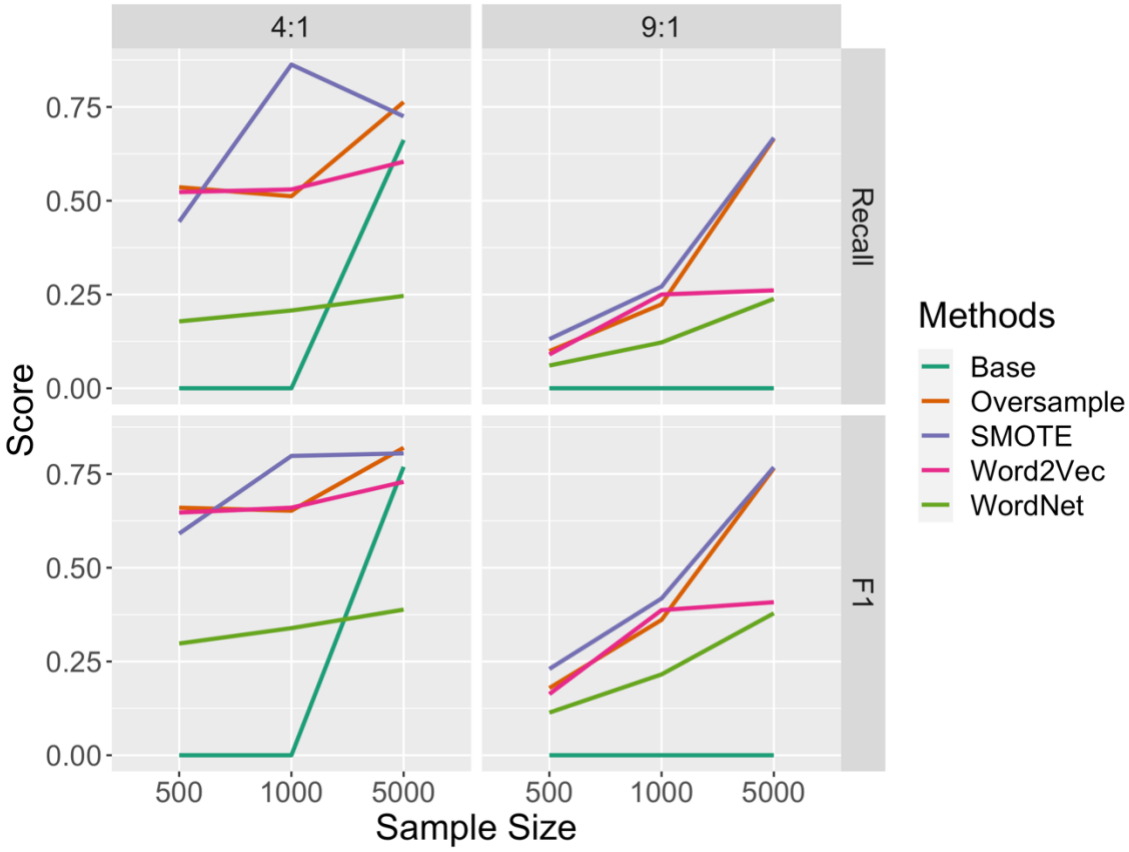


*Figure 11 Recalls and F1 Scores of the Fully Connected Neural Network with Different Methods for Imbalanced Data Trained with the IMDB Reviews Dataset*

The performance of the base LSTM model is better than that of logistic regression and the fully connected neural network. However, the oversampling method does not help the LSTM model this time; it even worsens its performance. Instead, Word2Vec augmentation gears up the performance of LSTM and achieves the best results on the

recall and F1 score in data of all sample sizes and imbalance ratios (see table 7 and figure 12).

| Imbalance Ratio | Data size | Methods | Accuracy | F1 | Precision | Recall |
|---|---|---|---|---|---|---|
| 4:1 | 500 | Base | 0.585 | 0.385 | 0.773 | 0.256 |
| | | Oversample | 0.536 | 0.196 | 0.805 | 0.111 |
| | | Word2Vec | 0.619 | 0.540 | 0.698 | 0.440 |
| | | WordNet | 0.560 | 0.351 | 0.696 | 0.235 |
| | 1000 | Base | 0.619 | 0.493 | 0.757 | 0.366 |
| | | Oversample | 0.606 | 0.456 | 0.762 | 0.325 |
| | | Word2Vec | 0.687 | 0.643 | 0.762 | 0.556 |
| | | WordNet | 0.584 | 0.372 | 0.795 | 0.243 |
| | 5000 | Base | 0.744 | 0.701 | 0.860 | 0.591 |
| | | Oversample | 0.697 | 0.596 | 0.920 | 0.441 |
| | | Word2Vec | 0.759 | 0.714 | 0.899 | 0.592 |
| | | WordNet | 0.655 | 0.537 | 0.839 | 0.395 |
| 9:1 | 500 | Base | 0.532 | 0.224 | 0.700 | 0.133 |
| | | Oversample | 0.511 | 0.089 | 0.812 | 0.047 |
| | | Word2Vec | 0.556 | 0.328 | 0.706 | 0.214 |
| | | WordNet | 0.502 | 0.044 | 0.802 | 0.023 |
| | 1000 | Base | 0.545 | 0.217 | 0.850 | 0.124 |
| | | Oversample | 0.538 | 0.184 | 0.875 | 0.103 |
| | | Word2Vec | 0.576 | 0.404 | 0.704 | 0.283 |
| | | WordNet | 0.546 | 0.273 | 0.725 | 0.168 |
| | 5000 | Base | 0.602 | 0.377 | 0.916 | 0.237 |
| | | Oversample | 0.582 | 0.318 | 0.924 | 0.192 |
| | | Word2Vec | 0.645 | 0.479 | 0.935 | 0.322 |
| | | WordNet | 0.581 | 0.338 | 0.849 | 0.211 |

*Table 7 Results of LSTM with Different Methods for Imbalanced Data Trained with the IMDB Reviews Dataset (the highlight indicates the best recall and F1 score)*

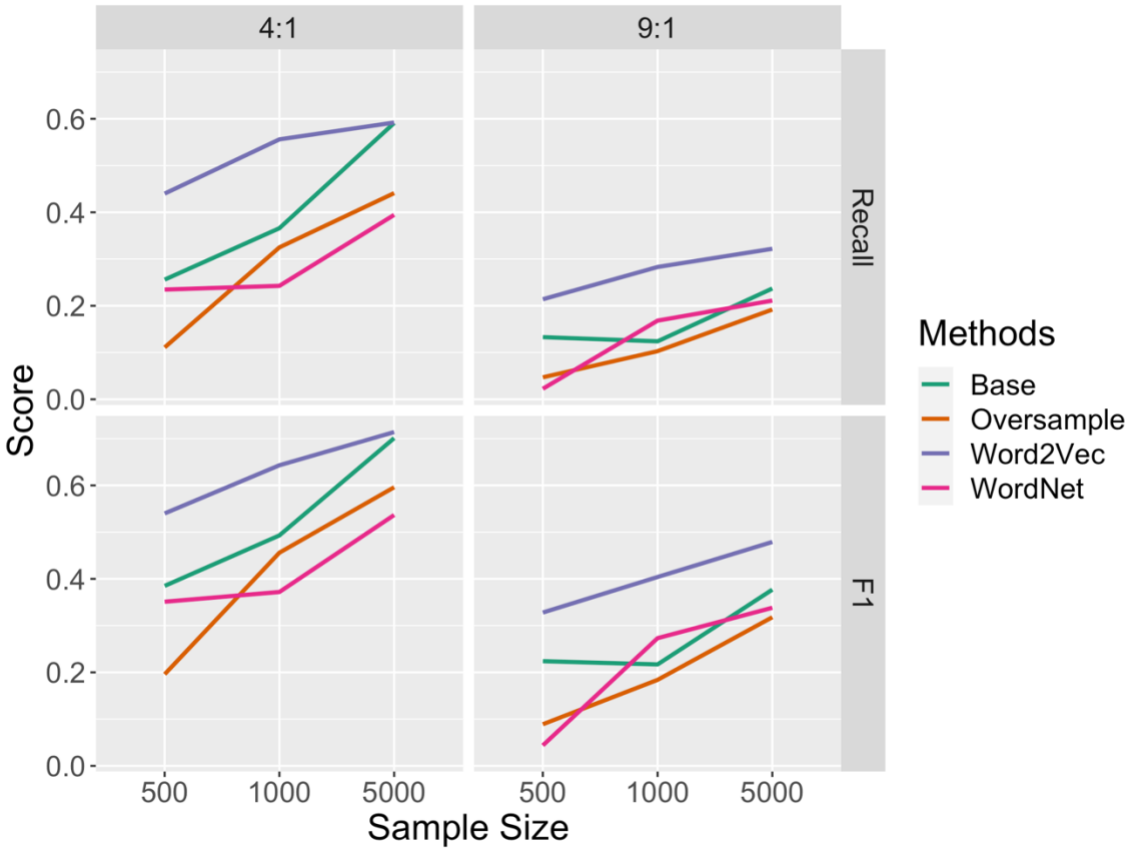


*Figure 12 Recalls and F1 Scores of LSTM with Different Methods for Imbalanced Data Trained with the IMDB Reviews Dataset*

The Word2Vec augmentation method also achieves the best performance with BERT fine-tuning (see table 8 and figure 13). When the imbalance ratio is 4:1, its F1 scores are consistently over 88%, and the recalls are over 84%, regardless of the sample size. When the imbalance ratio is 9:1, it is more sensitive to the sample size but still achieves a 70.7% F1 score and 56.6% recall with only 500 training data.

| Imbalance Ratio | Data size | Methods | Accuracy | F1 | Precision | Recall |
|---|---|---|---|---|---|---|
| 4:1 | 500 | Base | 0.708 | 0.615 | 0.924 | 0.462 |
| | | Oversample | 0.878 | 0.873 | 0.926 | 0.826 |
| | | Word2Vec | 0.885 | 0.885 | 0.898 | 0.873 |
| | | WordNet | 0.778 | 0.738 | 0.920 | 0.616 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 1000 | Base | 0.814 | 0.781 | 0.967 | 0.655 |
| | | Oversample | 0.882 | 0.876 | 0.939 | 0.822 |
| | | Word2Vec | 0.896 | 0.895 | 0.923 | 0.868 |
| | | WordNet | 0.846 | 0.826 | 0.961 | 0.725 |
| | 5000 | Base | 0.887 | 0.879 | 0.962 | 0.809 |
| | | Oversample | 0.892 | 0.886 | 0.956 | 0.825 |
| | | Word2Vec | 0.902 | 0.897 | 0.958 | 0.843 |
| | | WordNet | 0.897 | 0.892 | 0.953 | 0.839 |
| 9:1 | 500 | Base | 0.623 | 0.410 | 0.990 | 0.259 |
| | | Oversample | 0.709 | 0.607 | 0.963 | 0.443 |
| | | Word2Vec | 0.761 | 0.707 | 0.938 | 0.566 |
| | | WordNet | 0.563 | 0.248 | 0.977 | 0.142 |
| | 1000 | Base | 0.683 | 0.548 | 0.990 | 0.379 |
| | | Oversample | 0.761 | 0.697 | 0.977 | 0.542 |
| | | Word2Vec | 0.805 | 0.766 | 0.977 | 0.630 |
| | | WordNet | 0.759 | 0.694 | 0.974 | 0.539 |
| | 5000 | Base | 0.838 | 0.814 | 0.978 | 0.697 |
| | | Oversample | 0.826 | 0.797 | 0.980 | 0.671 |
| | | Word2Vec | 0.859 | 0.842 | 0.974 | 0.742 |
| | | WordNet | 0.809 | 0.772 | 0.979 | 0.637 |

*Table 8 Results of BERT Fine-Tuning with Different Methods for Imbalanced Data Trained with the IMDB Reviews Dataset (the highlight indicates the best recall and F1 score)*

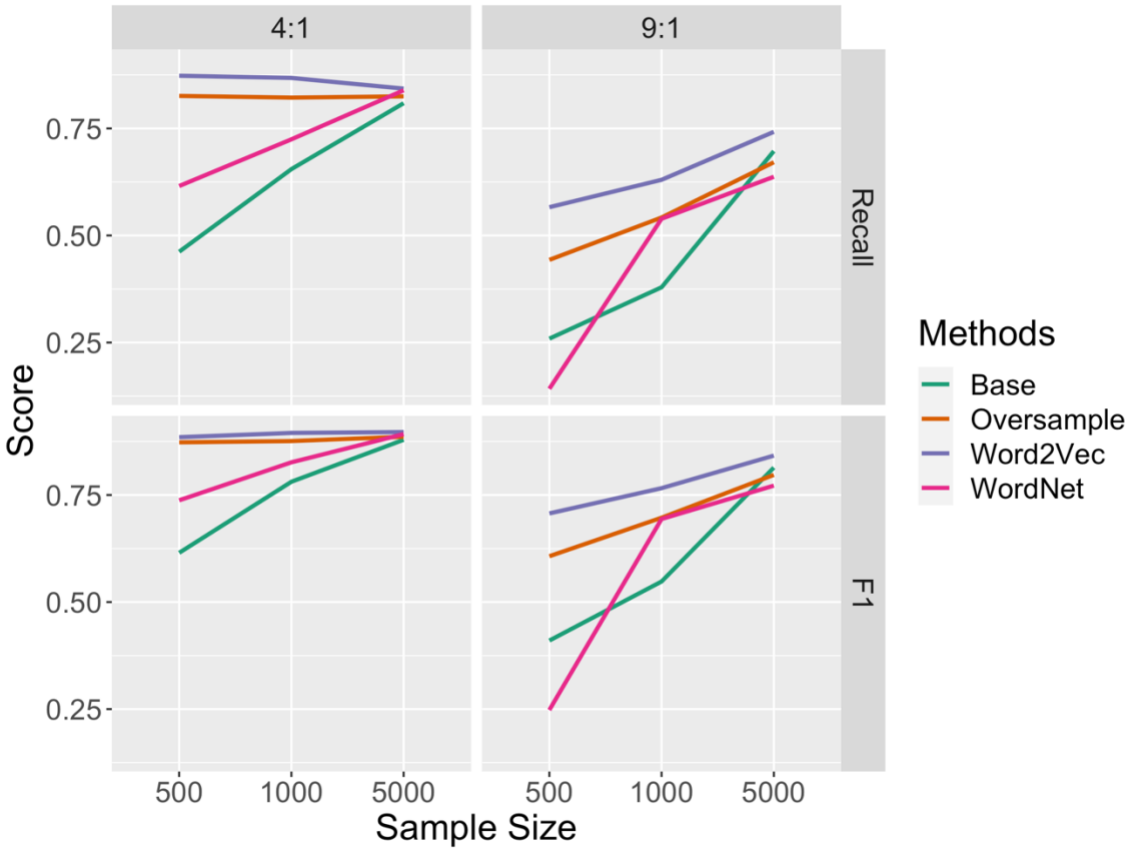


*Figure 13 Recalls and F1 Scores of BERT Fine-Tuning with Different Methods for Imbalanced Data Trained with the IMDB Reviews Dataset*

Comparing the best performance generated by the methods for imbalanced data across models, we find that BERT fine-tuning with Word2Vec augmentation consistently outperforms other models (see figure 14), especially when the data size is small (e.g., 500) and highly imbalanced (e.g., 9:1). When the data size is relatively large (e.g., 5,000), BERT fine-tuning with Word2Vec augmentation still performs the best, but the distance between its performance and other models is smaller.

It should be noted that logistic regression with oversampling outperforms the FC-NN and LSTM models. This might be attributed to the small data sizes in the experiments, where simple models are preferred. Although BERT is a complex deep

learning model, it benefits from pre-training and transfer learning. Even without Word2Vec augmentation, BERT still achieves similar or even better results than the best results of other models (see figure 14).

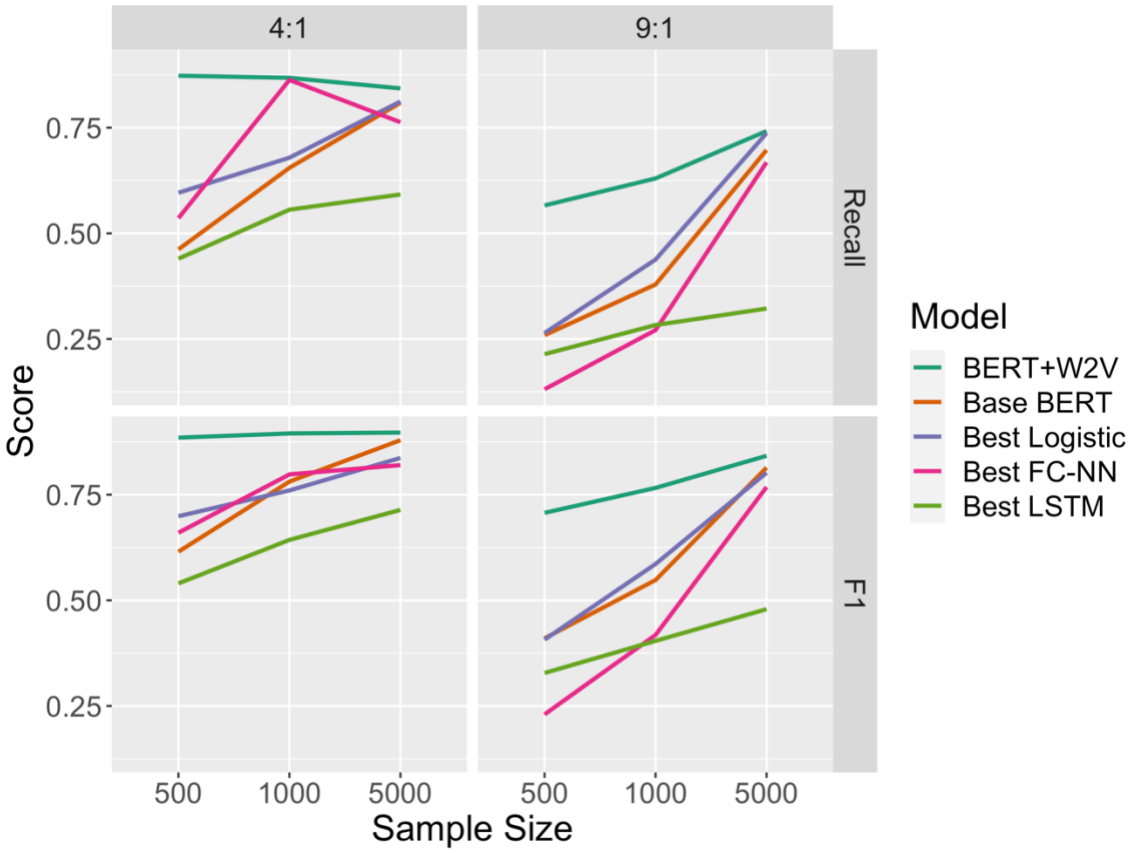


*Figure 14 Best Recalls and F1 Scores across Models Trained with the IMDB Reviews Dataset*

Another finding is that Word2Vec augmentation does not go well with simple models. In logistic regression, Word2Vec augmentation has the worst performance compared to other methods for imbalanced data. In fully connected neural networks, Word2Vec augmentation's performance improves and becomes closer to other methods. In LSTM and BERT, Word2Vec augmentation outperforms other methods. This phenomenon might be because Word2Vec augmentation largely increases the variance of

data (as shown before, the documents created by Word2Vec augmentation can be very different from the original documents), and therefore needs more complex models to handle the generated data.

Word2Vec augmentation can also influence the learning speed of deep learning models. Figure 15 shows that without methods for imbalanced data, the BERT model did not improve in the first four epochs of training. The oversampling methods helped the model learn faster but made it soon overfitted the training data after the third epoch. Word2Vec augmentation also increased the learning speed but delayed the overfitting to around the fifth epoch.

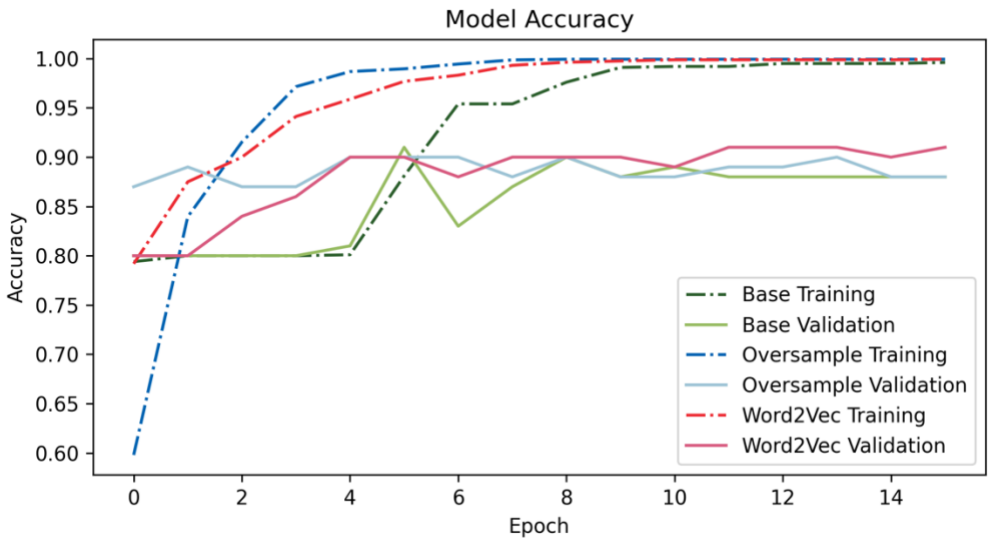


*Figure 15 Training History of BERT Model with the IMDB Dataset of Size 1000 and Imbalance Ratio 4:1.*

### *4.3.2. Toxic Comments Data Results*

The Toxic Comments dataset is different from IMDB Reviews. Its testing set is imbalanced at the ratio of 9:1. Unlike the testing results in the IMDB dataset, where a high F1 score is associated with a high recall value, a high F1 score does not necessarily

mean a high recall value in the Toxic dataset. This is because the F1 score is calculated as $\frac{TP}{TP+\frac{1}{2}(FP+FN)}$, where TP is the true positive, FP is the false positive, and FN is the false negative. If the positive class is the minority, focusing on identifying the positive class is more likely to generate FP than the chance of generating FN by focusing on identifying the majority. Therefore, this study mainly looks at recall values when evaluating the models' performance in the Toxic dataset.

The recall performance of models and methods for imbalanced data on the Toxic Comments dataset is slightly different from that of the IMDB Reviews dataset. In logistic regression, the oversampling method and SMOTE produce the best recall values three times, respectively, while the best F1 scores were most frequently produced by the oversampling method (see table 9 and figure 16). The pattern is similar in FC-NN, where SMOTE is more likely to generate the best recall values while the oversampling method is more likely to yield the best F1 scores (see table 10 and figure 17).

| Imbalance Ratio | Data size | Methods | Accuracy | F1 | Precision | Recall |
|---|---|---|---|---|---|---|
| 4:1 | 500 | Base | 0.915 | 0.246 | 0.898 | 0.143 |
| | | Oversample | 0.880 | 0.469 | 0.410 | 0.548 |
| | | Word2Vec | 0.880 | 0.440 | 0.400 | 0.488 |
| | | WordNet | 0.904 | 0.447 | 0.505 | 0.401 |
| | | SMOTE | 0.776 | 0.360 | 0.249 | 0.648 |
| | | Boosting | 0.863 | 0.351 | 0.325 | 0.381 |
| | 1000 | Base | 0.923 | 0.386 | 0.843 | 0.250 |
| | | Oversample | 0.870 | 0.499 | 0.397 | 0.669 |
| | | Word2Vec | 0.882 | 0.464 | 0.414 | 0.527 |
| | | WordNet | 0.907 | 0.499 | 0.522 | 0.478 |
| | | SMOTE | 0.797 | 0.414 | 0.287 | 0.740 |
| | | Boosting | 0.899 | 0.501 | 0.479 | 0.525 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 5000 | Base | 0.927 | 0.562 | 0.667 | 0.486 |
| | | Oversample | 0.853 | 0.511 | 0.377 | 0.795 |
| | | Word2Vec | 0.888 | 0.505 | 0.441 | 0.592 |
| | | WordNet | 0.908 | 0.558 | 0.523 | 0.598 |
| | | SMOTE | 0.796 | 0.426 | 0.293 | 0.782 |
| | | Boosting | 0.910 | 0.554 | 0.535 | 0.574 |
| 9:1 | 500 | Base | 0.905 | 0.030 | 1.000 | 0.015 |
| | | Oversample | 0.906 | 0.482 | 0.517 | 0.451 |
| | | Word2Vec | 0.895 | 0.423 | 0.451 | 0.398 |
| | | WordNet | 0.913 | 0.358 | 0.625 | 0.251 |
| | | SMOTE | 0.867 | 0.365 | 0.339 | 0.396 |
| | | Boosting | 0.885 | 0.338 | 0.380 | 0.304 |
| | 1000 | Base | 0.910 | 0.134 | 0.966 | 0.072 |
| | | Oversample | 0.892 | 0.493 | 0.451 | 0.543 |
| | | Word2Vec | 0.894 | 0.440 | 0.453 | 0.428 |
| | | WordNet | 0.912 | 0.427 | 0.582 | 0.337 |
| | | SMOTE | 0.821 | 0.413 | 0.303 | 0.650 |
| | | Boosting | 0.906 | 0.440 | 0.522 | 0.380 |
| | 5000 | Base | 0.925 | 0.422 | 0.836 | 0.282 |
| | | Oversample | 0.864 | 0.512 | 0.392 | 0.735 |
| | | Word2Vec | 0.898 | 0.489 | 0.477 | 0.502 |
| | | WordNet | 0.912 | 0.522 | 0.553 | 0.494 |
| | | SMOTE | 0.795 | 0.404 | 0.281 | 0.718 |
| | | Boosting | 0.918 | 0.481 | 0.627 | 0.390 |

*Table 9 Results of Logistic Regression with Different Methods for Imbalanced Data Trained with the Toxic Comments Dataset (the yellow highlight indicates the best recall, and the green highlight indicates the best F1 score)*

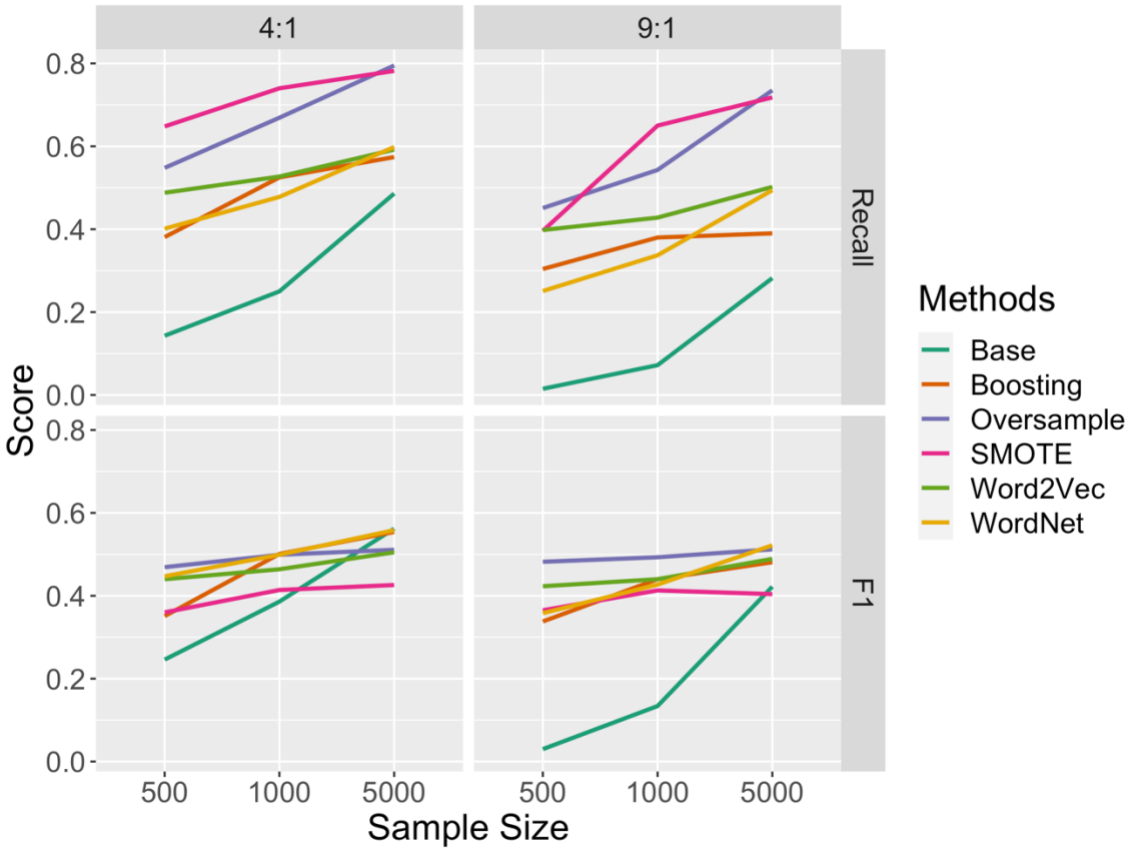


*Figure 16 Recalls and F1 Scores of Logistic Regression with Different Methods for Imbalanced Data Trained with the Toxic Comments Dataset*

| Imbalance Ratio | Data size | Methods | Accuracy | F1 | Precision | Recall |
|---|---|---|---|---|---|---|
| 4:1 | 500 | Base | 0.903 | 0.000 | - | 0.000 |
| | | Oversample | 0.892 | 0.464 | 0.447 | 0.484 |
| | | Word2Vec | 0.834 | 0.411 | 0.313 | 0.596 |
| | | WordNet | 0.901 | 0.440 | 0.487 | 0.402 |
| | | SMOTE | 0.829 | 0.413 | 0.310 | 0.621 |
| | 1000 | Base | 0.919 | 0.456 | 0.655 | 0.350 |
| | | Oversample | 0.887 | 0.499 | 0.438 | 0.579 |
| | | Word2Vec | 0.839 | 0.428 | 0.327 | 0.620 |
| | | WordNet | 0.891 | 0.456 | 0.441 | 0.472 |
| | | SMOTE | 0.829 | 0.435 | 0.320 | 0.679 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 5000 | Base | 0.876 | 0.505 | 0.413 | 0.650 |
| | | Oversample | 0.887 | 0.568 | 0.450 | 0.767 |
| | | Word2Vec | 0.823 | 0.436 | 0.315 | 0.708 |
| | | WordNet | 0.907 | 0.551 | 0.518 | 0.589 |
| | | SMOTE | 0.822 | 0.457 | 0.325 | 0.771 |
| 9:1 | 500 | Base | 0.903 | 0.000 | - | 0.000 |
| | | Oversample | 0.908 | 0.110 | 0.946 | 0.058 |
| | | Word2Vec | 0.901 | 0.401 | 0.487 | 0.342 |
| | | WordNet | 0.911 | 0.350 | 0.594 | 0.248 |
| | | SMOTE | 0.905 | 0.373 | 0.522 | 0.290 |
| | 1000 | Base | 0.903 | 0.000 | - | 0.000 |
| | | Oversample | 0.912 | 0.454 | 0.569 | 0.378 |
| | | Word2Vec | 0.890 | 0.431 | 0.432 | 0.430 |
| | | WordNet | 0.908 | 0.427 | 0.537 | 0.354 |
| | | SMOTE | 0.856 | 0.423 | 0.346 | 0.542 |
| | 5000 | Base | 0.903 | 0.000 | - | 0.000 |
| | | Oversample | 0.888 | 0.540 | 0.448 | 0.679 |
| | | Word2Vec | 0.864 | 0.423 | 0.359 | 0.516 |
| | | WordNet | 0.905 | 0.520 | 0.510 | 0.530 |
| | | SMOTE | 0.850 | 0.456 | 0.352 | 0.647 |

*Table 10 Results of the Fully Connected Neural Network with Different Methods for Imbalanced Data Trained with the Toxic Comments Dataset (the yellow highlight indicates the best recall, and the green highlight indicates the best F1 score)*

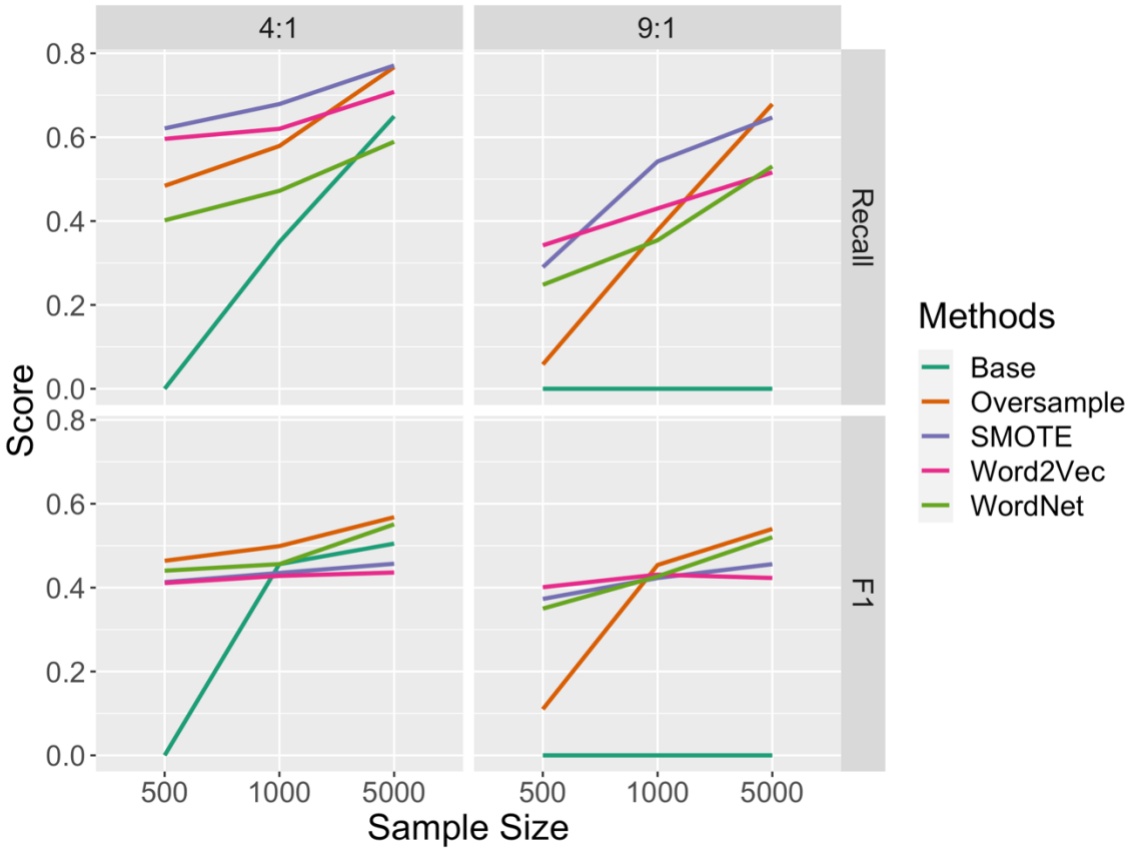


*Figure 17 Recalls and F1 Scores of the Fully Connected Neural Network with Different Methods for Imbalanced Data Trained with the Toxic Comments Dataset*

In the LSTM models, the Word2Vec augmentation method most frequently yields the best recall and F1 scores, similar to the IMDB dataset results (see table 11 and figure 18).

| Imbalance Ratio | Data size | Methods | Accuracy | F1 | Precision | Recall |
|---|---|---|---|---|---|---|
| 4:1 | 500 | Base | 0.863 | 0.406 | 0.351 | 0.481 |
| | | Oversample | 0.878 | 0.383 | 0.377 | 0.390 |
| | | Word2Vec | 0.843 | 0.388 | 0.312 | 0.512 |
| | | WordNet | 0.843 | 0.381 | 0.309 | 0.498 |
| | 1000 | Base | 0.860 | 0.427 | 0.355 | 0.536 |
| | | Oversample | 0.878 | 0.441 | 0.396 | 0.497 |
| | | Word2Vec | 0.819 | 0.398 | 0.294 | 0.620 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | WordNet | 0.875 | 0.457 | 0.394 | 0.543 |
| | 5000 | Base | 0.873 | 0.478 | 0.397 | 0.599 |
| | | Oversample | 0.828 | 0.473 | 0.337 | 0.795 |
| | | Word2Vec | 0.888 | 0.505 | 0.443 | 0.586 |
| | | WordNet | 0.872 | 0.536 | 0.413 | 0.761 |
| 9:1 | 500 | Base | 0.903 | 0.000 | - | 0.000 |
| | | Oversample | 0.883 | 0.336 | 0.372 | 0.305 |
| | | Word2Vec | 0.873 | 0.384 | 0.363 | 0.408 |
| | | WordNet | 0.872 | 0.314 | 0.327 | 0.302 |
| | 1000 | Base | 0.905 | 0.378 | 0.521 | 0.296 |
| | | Oversample | 0.906 | 0.334 | 0.534 | 0.243 |
| | | Word2Vec | 0.876 | 0.429 | 0.388 | 0.479 |
| | | WordNet | 0.849 | 0.363 | 0.307 | 0.444 |
| | 5000 | Base | 0.900 | 0.548 | 0.487 | 0.627 |
| | | Oversample | 0.893 | 0.523 | 0.462 | 0.603 |
| | | Word2Vec | 0.911 | 0.530 | 0.543 | 0.517 |
| | | WordNet | 0.918 | 0.520 | 0.600 | 0.460 |

*Table 11 Results of LSTM with Different Methods for Imbalanced Data Trained with the Toxic Comments Dataset (the yellow highlight indicates the best recall, and the green highlight indicates the best F1 score)*

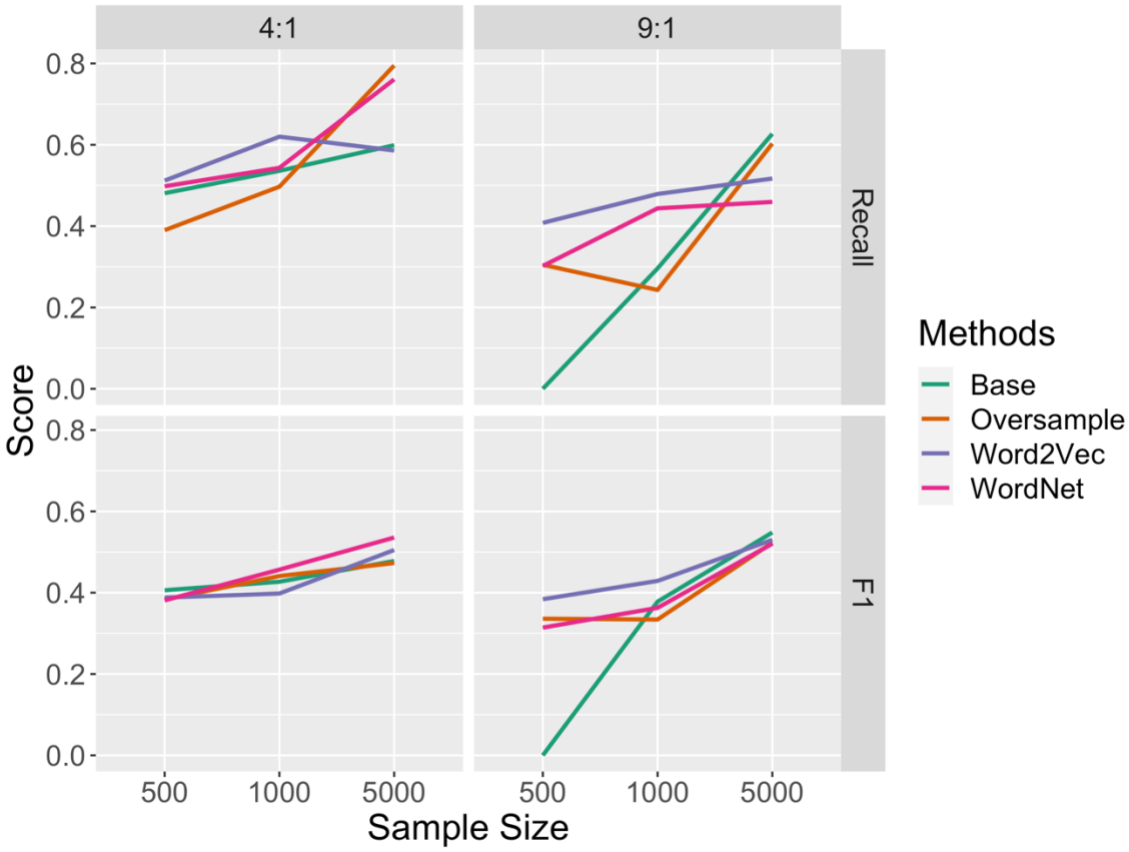


*Figure 18 Recalls and F1 Scores of LSTM with Different Methods for Imbalanced Data Trained with the Toxic Comments Dataset*

In the BERT model, different from the performance in IMDB data, the oversampling method and base model often generate the best F1 scores. However, this is mainly because the testing dataset is highly imbalanced. The Word2Vec augmentation method consistently generates the best recall scores, especially when the imbalance ratio is 9:1, indicating that Word2Vec augmentation can increase the model's ability to identify the minority class (see table 12 and figure 19). This study also randomly selected a balanced testing dataset (6,028 toxic comments and 6,028 non-toxic comments) and found that the Word2Vec augmentation method produced the best F1 scores most frequently with this testing data due to its better recall performance (see figure 20). The

confusion matrices (figure 21 left) show that even after oversampling, the BERT model still tends to classify more documents into the majority class. This tendency is lowered after the Word2Vec augmentation (figure 21 right), demonstrating that the Word2Vec augmentation method can better help models recognize the minority class than the oversampling method.

| Imbalance Ratio | Data size | Methods | Accuracy | F1 | Precision | Recall |
|---|---|---|---|---|---|---|
| 4:1 | 500 | Base | 0.902 | 0.583 | 0.512 | 0.677 |
| | | Oversample | 0.883 | 0.585 | 0.458 | 0.808 |
| | | Word2Vec | 0.862 | 0.515 | 0.401 | 0.720 |
| | | WordNet | 0.895 | 0.511 | 0.487 | 0.539 |
| | 1000 | Base | 0.884 | 0.582 | 0.459 | 0.796 |
| | | Oversample | 0.882 | 0.587 | 0.456 | 0.822 |
| | | Word2Vec | 0.861 | 0.552 | 0.411 | 0.840 |
| | | WordNet | 0.896 | 0.591 | 0.494 | 0.736 |
| | 5000 | Base | 0.902 | 0.649 | 0.510 | 0.890 |
| | | Oversample | 0.902 | 0.643 | 0.510 | 0.868 |
| | | Word2Vec | 0.884 | 0.611 | 0.465 | 0.891 |
| | | WordNet | 0.894 | 0.631 | 0.490 | 0.886 |
| 9:1 | 500 | Base | 0.923 | 0.569 | 0.660 | 0.500 |
| | | Oversample | 0.918 | 0.541 | 0.629 | 0.474 |
| | | Word2Vec | 0.864 | 0.469 | 0.389 | 0.589 |
| | | WordNet | 0.890 | 0.393 | 0.447 | 0.351 |
| | 1000 | Base | 0.921 | 0.576 | 0.635 | 0.528 |
| | | Oversample | 0.914 | 0.591 | 0.570 | 0.615 |
| | | Word2Vec | 0.890 | 0.572 | 0.473 | 0.724 |
| | | WordNet | 0.908 | 0.565 | 0.545 | 0.586 |
| | 5000 | Base | 0.928 | 0.677 | 0.623 | 0.742 |
| | | Oversample | 0.925 | 0.673 | 0.602 | 0.763 |
| | | Word2Vec | 0.897 | 0.622 | 0.495 | 0.835 |
| | | WordNet | 0.919 | 0.659 | 0.576 | 0.768 |

*Table 12 Results of BERT Fine-Tuning with Different Methods for Imbalanced Data Trained with the Toxic Comments Dataset (the green highlight indicates the best recall, and the yellow highlight indicates the best F1 score)*

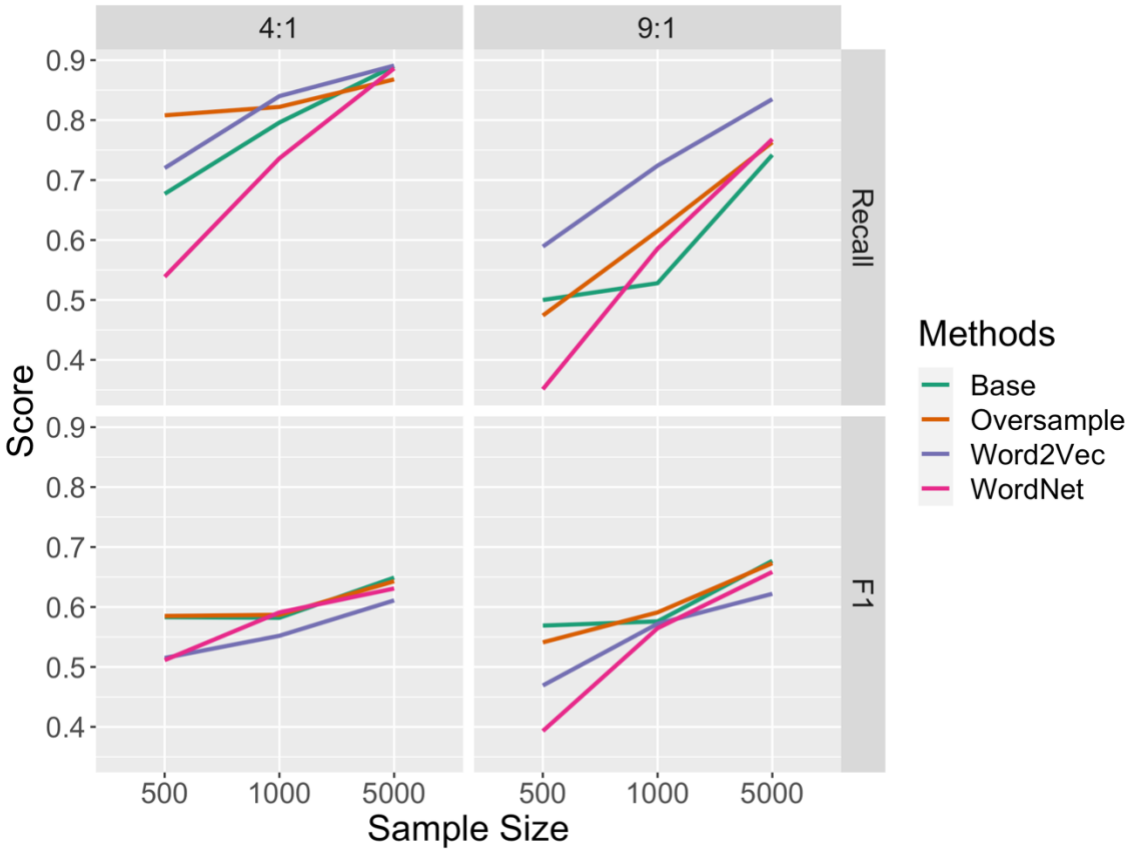


*Figure 19 Recalls and F1 Scores of BERT Fine-Tuning with Different Methods for Imbalanced Data Trained with the Toxic Comments Dataset*

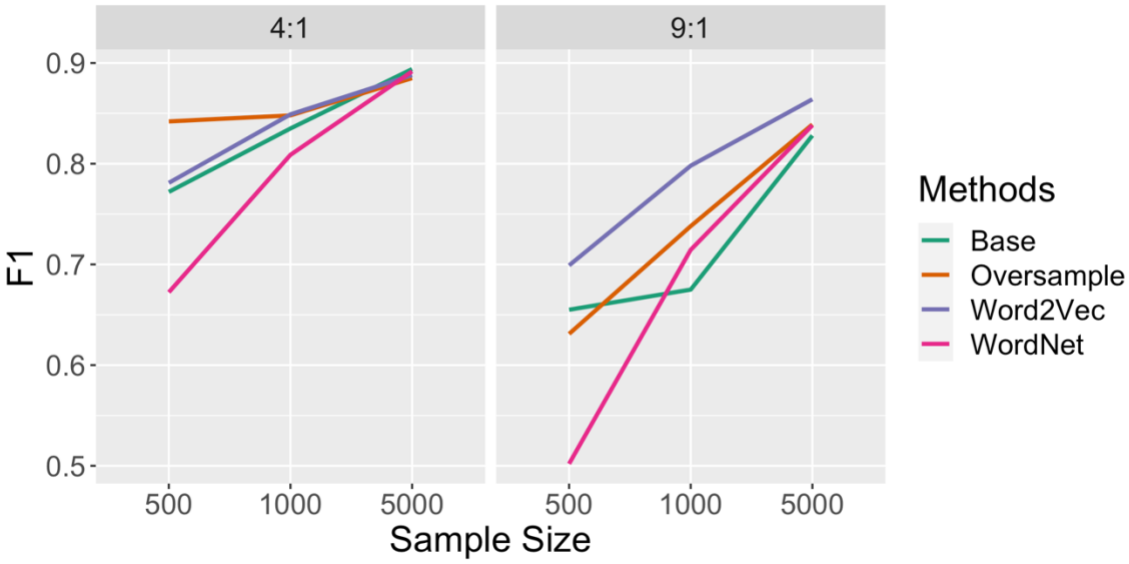


*Figure 20 F1 Scores of BERT Fine-Tuning with Different Methods for Imbalanced Data on Balanced Testing data Trained with the Toxic Comments Dataset*

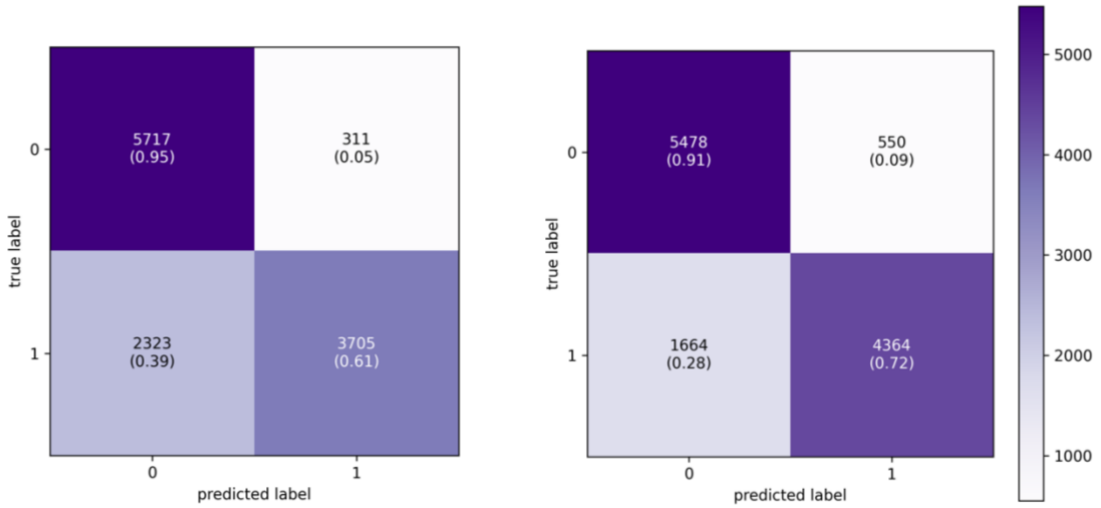


*Figure 21 Confusion Matrices of BERT Fine-Tuning with Oversampling (Left) and Word2Vec Augmentation (Right) Methods on the Balanced Testing Data Trained with the Toxic Comments Dataset of Size 1,000 and Imbalance Ratio 9:1.*

Again, the results indicate that the BERT model with Word2Vec augmentation always outperform other models and methods(see figure 22).

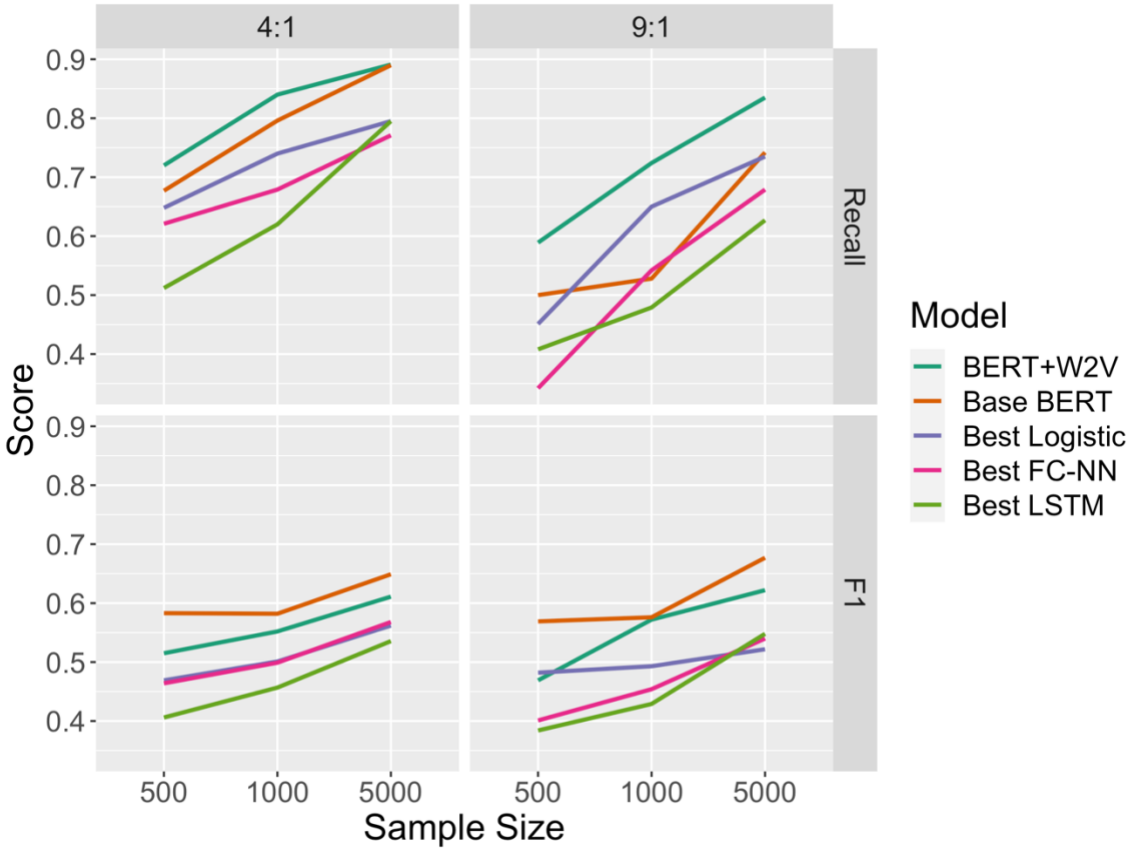


*Figure 22 Best Recalls and F1 Scores across Models Trained with the Toxic Comments Dataset*

However, it should be noticed that even when the testing dataset is balanced, the performance improvement generated by the Word2Vec augmentation method is not as significant as it with the IMDB dataset, especially when the imbalance ratio is low (e.g., 4:1). One reason might be that documents in the Toxic Comments dataset are much shorter than those in the IMDB dataset (see figures 7 and 9). Replacing words in a shorter document may create more noise than in a longer document and therefore does not generate ideal improvements.

## 5. Conclusion and Future Directions

Social science has a long tradition of analyzing text messages, which provides machine learning algorithms with valuable training resources. However, the text data annotated by social scientists are usually small-sized and imbalanced in categories, limiting its application with machine learning and deep learning models. Therefore, this study proposes a pipeline to combine Word2Vec augmentation with BERT and examines its effectiveness along with other three methods for imbalanced data (i.e., boosting, SMOTE, and oversampling) and three machine learning models (i.e., logistic regression, fully connected neural network, and LSTM) on datasets of various sizes (e.g., 500, 1,000, and 5,000 training documents) and imbalance ratios (e.g., 4:1 and 9:1).

The results show that, generally, the oversampling method generates the best performance in terms of recalls and F1 scores for simple classifiers such as logistic regression. For fully connected neural networks, SMOTE usually produces the best results, followed by oversampling methods. For LSTM and BERT fine-tuning, the Word2Vec augmentation method yields the best performance. Although BERT fine-tuning almost consistently outperforms other models regardless of whether having any methods for imbalanced data, the Word2Vec augmentation method helps it achieve better performance, especially better recalls. This improvement is particularly significant when the data size is small (e.g., 500 training documents) and highly imbalanced (e.g., 9:1). When the data size is large (e.g., over 5,000) or the imbalance ratio is not high (e.g., lower than 4:1), the improvement generated by the Word2Vec augmentation method becomes insignificant; instead, the BERT model itself has a decent performance.

This study contributes to the field mainly from three perspectives. First, it successfully develops a promising solution—combining Word2Vec augmentation with BERT—for small-sized, imbalanced text classification tasks, which can benefit scholars, especially social science scholars, to utilize the resources from content analysis to develop machine learning classifiers to increase the generalizability of their studies. Second, this study examines how sample sizes and imbalance ratios influence the effectiveness of Word2Vec augmentation with BERT and reveals the limitations of the method under certain conditions, responding to the previous studies' [e.g., 12, 34] suspiciousness about the efficacy of text augmentation and words replacement methods with context-based models. Furthermore, this study compares different methods for imbalanced data with machine learning models and tests their performance under different conditions. Some interesting patterns were detected, such as Word2Vec augmentation does not work well with simple models.

In addition, this study could enlighten future research from two aspects. First, this study found that, in general, the Word2Vec augmentation method has a more significant effect on the BERT model with the IMDB dataset than with the Toxic dataset. One possible reason might be that the Word2Vec augmentation method works better with longer documents. Future studies could examine how the document length influences the performance of the Word2Vec augmentation method. Second, in this study, the Word2Vec augmentation method randomly replaces some words in documents with words of high similarities in its corpus. Future studies could optimize the selections of words for the replacement (for example, only replacing the words with high or low weights in the classification process) and test their effects.

## 6. References

[1] Y. R. Tausczik and J. W. Pennebaker, "The psychological meaning of words: LIWC and computerized text analysis methods," *Journal of Language and Social Psychology*, vol. 29, no. 1, pp. 24–54, 2010, doi: 10.1177/0261927X09351676.

[2] K. Welbers, W. Van Atteveldt, and K. Benoit, "Text analysis in R," *Communication Methods and Measures*, vol. 11, no. 4, pp. 245-265, 2017.

[3] R. Sathya and A. Abraham, "Comparison of supervised and unsupervised learning algorithms for pattern classification," *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 2, pp. 34-38, 2013.

[4] K. Li, D. Yan, Y. Liu, and Q. Zhu, "A network-based feature extraction model for imbalanced text data," *Expert Systems with Applications*, vol. 195, p. 116600, 2022.

[5] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221-232, 2016.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[7] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *European conference on principles of data mining and knowledge discovery*, 2003: Springer, pp. 107-119.

[8] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary,"

*Journal of artificial intelligence research,* vol. 61, pp. 863–905, 2018, doi: 10.1613/jair.1.11192.

[9] F. Ren and J. Deng, "Background knowledge based multi-stream neural network for text classification," *Applied Sciences,* vol. 8, no. 12, p. 2472, 2018.

[10] Y. Li, H. Guo, Q. Zhang, M. Gu, and J. Yang, "Imbalanced text sentiment classification using universal and domain-specific knowledge," *Knowledge-Based Systems,* vol. 160, pp. 1-15, 2018.

[11] M. Okkalioglu and B. D. Okkalioglu, "AFE-MERT: imbalanced text classification with abstract feature extraction," *Applied Intelligence,* pp. 1-17, 2022.

[12] H. T. Madabushi, E. Kochkina, and M. Castelle, "Cost-sensitive BERT for generalisable sentence classification with imbalanced data," *arXiv preprint arXiv:2003.11563,* 2020.

[13] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognition,* vol. 40, no. 12, pp. 3358-3378, 2007.

[14] C. Padurariu and M. E. Breaban, "Dealing with data imbalance in text classification," *Procedia Computer Science,* vol. 159, pp. 736-745, 2019.

[15] U. R. Salunkhe and S. N. Mali, "Classifier ensemble design for imbalanced data classification: a hybrid approach," *Procedia Computer Science,* vol. 85, pp. 725-732, 2016.

[16] N. Japkowicz, "The class imbalance problem: Significance and strategies," in *Proceedings of the 2000 International Conference on Artificial Intelligence*, Las Vegas, Nevada, 2000, vol. 56: Citeseer.

[17] T. Chen, R. Xu, B. Liu, Q. Lu, and J. Xu, "WEMOTE-Word Embedding based Minority Oversampling Technique for Imbalanced Emotion and Sentiment Classification," in *Workshop on Issues of Sentiment Discovery and Opinion Mining*, 2014.

[18] E. L. Iglesias, A. S. Vieira, and L. Borrajo, "An HMM-based over-sampling technique to improve text classification," *Expert Systems with Applications,* vol. 40, no. 18, pp. 7184-7192, 2013.

[19] S. Wang, D. Li, L. Zhao, and J. Zhang, "Sample cutting method for imbalanced text sentiment classification based on BRC," *Knowledge-Based Systems,* vol. 37, pp. 451-461, 2013.

[20] H. Ogura, H. Amano, and M. Kondo, "Comparison of metrics for feature selection in imbalanced text classification," *Expert Systems with Applications,* vol. 38, no. 5, pp. 4978-4989, 2011.

[21] F. Ren and M. G. Sohrab, "Class-indexing-based term weighting for automatic text classification," *Information Sciences,* vol. 236, pp. 109-125, 2013.

[22] B. Naderalvojoud, E. A. Sezer, and A. Ucan, "Imbalanced text categorization based on positive and negative term weighting approach," in *International Conference on Text, Speech, and Dialogue*, 2015: Springer, pp. 325-333.

[23] V. Marivate and T. Sefara, "Improving short text classification through global augmentation methods," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, 2020: Springer, pp. 385-399.

[24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint*, vol. 1301.3781v3, pp. 1-12, 2013.

[26] G. A. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39-41, 1995.

[27] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[28] S. Sharifirad, B. Jafarpour, and S. Matwin, "Boosting text classification performance on sexist tweets by text augmentation and text generation using a combination of knowledge graphs," in *Proceedings of the 2nd workshop on abusive language online (ALW2)*, 2018, pp. 107-114.

[29] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *Advances in neural information processing systems*, vol. 28, 2015.

[30] M. Aiken and M. Park, "The efficacy of round-trip translation for MT evaluation," *Translation Journal*, vol. 14, no. 1, pp. 1-10, 2010.

[31] X. Tang, H. Mou, J. Liu, and X. Du, "Research on automatic labeling of imbalanced texts of customer complaints based on text enhancement and layer-by-layer semantic matching," *Scientific Reports,* vol. 11, no. 1, pp. 1-11, 2021.

[32] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," *arXiv preprint arXiv:1511.06709,* 2015.

[33] S. T. Aroyehun and A. Gelbukh, "Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, 2018, pp. 90-97.

[34] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," *arXiv preprint arXiv:1901.11196,* 2019.

[35] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems: First International Workshop*, Cagliari, Italy, J. Kittler and F. Roli, Eds., 2000: Springer, pp. 1-15.

[36] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.

[37] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class adaboost," *Statistics and its Interface,* vol. 2, no. 3, pp. 349-360, 2009.

[38] A. Vezhnevets and V. Vezhnevets, "Modest AdaBoost-teaching AdaBoost to generalize better," in *Graphicon*, 2005, vol. 12, no. 5, pp. 987-997.

[39] T. Kudo and Y. Matsumoto, "A boosting algorithm for classification of semi-structured text," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, 2004, pp. 301-308.

[40] C. Ehrentraut, M. Ekholm, H. Tanushi, J. Tiedemann, and H. Dalianis, "Detecting hospital-acquired infections: a document classification approach using support vector machines and gradient tree boosting," *Health informatics journal,* vol. 24, no. 1, pp. 24-42, 2018.

[41] V. S. Sheng and C. X. Ling, "Thresholding for making classifiers cost-sensitive," in *AAAI*, 2006, vol. 6, pp. 476-81.

[42] E. S. Olivas, J. D. M. Guerrero, M. Martinez-Sober, J. R. Magdalena-Benedito, and L. Serrano, *Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques: Algorithms, methods, and techniques*. IGI global, 2009.

[43] Z. Xiao, L. Wang, and J. Du, "Improving the performance of sentiment classification on imbalanced datasets with transfer learning," *IEEE Access,* vol. 7, pp. 28281-28290, 2019.

[44] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint* vol. 1810.04805, 2018.

[45] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research,* vol. 12, no. Oct, pp. 2825-2830, 2011.

[46] *Keras*. (2015). Github. [Online]. Available: https://keras.io

[47] T. Wolf *et al.*, "Transformers: State-of-the-art natural language processing," *ArXiv,* vol. abs/1910.03771, 2019.

[48] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[49] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Thirteenth International Conference on Machine Learning*, Bari, Italy, L. Saitta, Ed., 1996: Morgan Kaufmann, pp. 148–156.

[50] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 559–563, 2017.

[51] M. Miháltz. Word2vec Google News Vectors [Online] Available: https://github.com/mmihaltz/word2vec-GoogleNews-vectors

[52] M. Ibrahim, M. Torki, and N. El-Makky, "Imbalanced toxic comments classification using data augmentation and deep learning," in *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, 2018: IEEE, pp. 875-878.

[53] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, 2011: Association for Computational Linguistics, pp. 142-150.

[54] Jigsaw. Toxic Comment Classification Challenge [Online] Available: https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/overview

## VITA

Lingshu Hu was born in Deyang, Sichuan, China. He obtained his Bachelor of Arts degree from the Communication University of Zhejiang in 2010, majoring in Broadcast Journalism. Then he went to the London School of Economics and Political Science for graduate studies and obtained his Master of Science degree in Gender, Media, and Culture. After graduation, he worked as a news editor and data journalist for four years at the *Guangzhou Daily* in China.

In 2017, Lingshu started his Ph.D. program at the University of Missouri's School of Journalism. His research focuses on computational methods, media analytics, social media, and the digital public. He applies various computational methods—such as machine learning and deep learning—to analyze patterns of communication and self-presentation in computer-mediated environments. During his Ph.D. studies, he earned a graduate certificate in AI and Machine Learning from the University of Missouri's computer science program and decided to further pursue a master's degree in computer science.

Lingshu Hu is currently an assistant professor of business administration and data science at Washington and Lee University.