applied sciences

MDPI

Article

Application of Generative Adversarial Networks and Shapley Algorithm Based on Easy Data Augmentation for Imbalanced Text Data

Jheng-Long Wu * and Shuoyen Huang

Department of Data Science, Soochow University, Taipei 111002, Taiwan
* Correspondence: jlwu@gm.scu.edu.tw

Abstract: Imbalanced data constitute an extensively studied problem in the field of machine learning classification because they result in poor training outcomes. Data augmentation is a method for increasing minority class diversity. In the field of text data augmentation, easy data augmentation (EDA) is used to generate additional data that would otherwise lack diversity and exhibit monotonic sentence patterns. Generative adversarial network (GAN) models can generate diverse sentence patterns by using the probability corresponding to each word in a language model. Therefore, hybrid EDA and GAN models can generate highly diverse and appropriate sentence patterns. This study proposes a hybrid framework that employs a generative adversarial network and Shapley algorithm based on easy data augmentation (HEGS) to improve classification performance. The experimental results reveal that the HEGS framework can generate highly diverse training sentences to form balanced text data and improve text classification performance for minority classes.

Keywords: generative adversarial network; easy data augmentation; text classification; text generation; imbalanced data

check for updates

Citation: Wu, J.-L.; Huang, S. Application of Generative Adversarial Networks and Shapley Algorithm Based on Easy Data Augmentation for Imbalanced Text Data. *Appl. Sci.* **2022**, *12*, 10964. https://doi.org/10.3390/app122110964

Academic Editor: Christos Bouras

Received: 15 September 2022
Accepted: 26 October 2022
Published: 29 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Imbalanced data result in poor training outcomes and thus constitute a key topic in natural language processing (NLP) [1,2]. Undersampling and oversampling are generally used to solve the problem of imbalanced data [3,4]; between the two methods, oversampling is more effective for text classification. The main solution applied in the oversampling method is data augmentation, where data on minority classes are augmented to obtain more similar samples. The easy data augmentation (EDA) method [5] was proposed as a text data augmentation method; it comprises the four processes of synonym replacement, random insertion, random deletion, and random swapping. Generative adversarial networks (GANs) [6] have also been used to generate data, such as sentimental text generation GAN (SentiGAN) [7] and category-aware GAN (CatGAN) [8]. These methods have performed well in a variety of applications. A GAN uses a generator and discriminator for adversarial training, where the generator generates data whose distribution approximates that of real data and the discriminator must distinguish between real data samples and generated samples. In addition, the Shapley algorithm [9,10] has been used as a metric to assess data quality, and it has been applied in many machine learning tasks. Specifically, the Shapley algorithm is used for data cleaning to achieve better training outcomes. In the field of NLP, the pretrained bidirectional encoder representations from transformers (BERT) model is becoming a popular classifier for downstream tasks involving finetuning training [11]. BERT incorporates pretraining for two tasks, namely, mask language modeling and next-sentence prediction, which can facilitate the learning of grammatical structures and semantic information [12].

Therefore, the present study proposes a hybrid framework to address the problem of imbalanced data, where a generative adversarial network and Shapley algorithm based

on easy data augmentation (HEGS) framework is used to obtain large sets of generated training data. The HEGS framework is based on the GAN model, the EDA method, and the Shapley algorithm to augment training sentences to obtain balanced data for classifier training. EDA generates an initial training sentence and uses it as the seed data for the next GAN-based model for models to be trained using numerous high-quality training sentences. The present study uses a GAN-based model such as generating sentimental texts via mixture adversarial networks (SentiGAN) to generate highly diverse sets of training sentences that are based on seed training data. Before classifier training is conducted for classification tasks, the BERT model is used as the proposed classifier to evaluate data augmentation performance.

The main contributions of the paper are as follows. (1) The GAN-based models obtain rich training data not only comprising original sentences of imbalanced data but also using the additional training sentences generated through EDA random mix process, which is different from other GAN training data in the past. The diverse initial training sentences can improve the GAN-based model to generate highly diverse sentences. (2) The application of the Shapley algorithm eliminates training sentences based on original training data and achieves high-quality and diverse training sentences. This Shapley algorithm successfully improves the performances of classifiers and remedies the imbalanced data in the downstream text classification task. (3) The proposed text augmentation method, HEGS hybrid framework, outperforms other single-text data augmentation methods.

The remainder of the present study is organized as follows: Section 2 presents a review of the literature on data augmentation, Section 3 describes the proposed HEGS framework, Section 4 reports the experimental results, and Section 5 discusses the findings and concludes the paper.

2. Related Work

In the following section, Section 2.1 provides an overview of text generation methods, and Section 2.2 introduces the concept of data augmentation for text classification.

2.1. Data Augmentation

Numerous studies have explored data augmentation methods such as easy data augmentation [5], where four straightforward processes are implemented to generate diverse data sets that improve classification performance. Adversarial attacks constitute another method for creating data. For example, the TextFooler method [13] generates text by identifying key words and replacing them with semantically and syntactically similar words to create new samples. The BAE method [14] uses a BERT model that incorporates pretraining masked language modeling to predict masked words and integrate with self-designed replacement, and insertion methods. The easy plug-in data augmentation (EPiDA) method [15] employs relative entropy maximization and conditional entropy maximization to evaluate the diversity and quality of generated samples. This method can also be combined with various data augmentation methods and classification models. The easier data augmentation (AEDA) method [16] uses punctuations as augmented data to solve the data scarcity problem. The text autoaugment (TAA) method [17] applies a self-designed algorithm to improve editing-based data augmentation methods. Specifically, TAA leverages compositional augmentation policies, each of which comprises three components. The first component is type, which includes random swapping, random deletion, term frequency–inverse document frequency (TF-IDF) insertion, TF-IDF replacement, and wordnet replace. The second component is probability, which is used to control the probability of usage. The third component is magnitude, which controls the portion of words that are changed. In summary, TAA enhances the generalization ability of a model and reduces the need for human annotation. Kobayashi [18] proposed the use of bidirectional long short-term memory (LSTM) to generate augmented data through input sentences and their labels. Wu et al. [19] performed BERT-masked language model pretraining tasks to randomly mask the words in sentences and output augmented data. Radford et al. [20]

proposed the generative pretrained transformer two (GPT-2) model, which can also be used to address problems with scarce data. For example, Anaby-Tavor et al. [21] proposed the language model based data augmentation (i.e., LAMBADA) method, which uses GPT-2, finetunes small sample data sets and generated samples, and selects generated samples through a classifier trained by trained data. Wu et al. [22] proposed a text smoothing method that uses BERT to encode one-hot representations into smoothed and interpolated representations for data augmentation. Jo et al. [23] proposed the DAGAM method, which uses three sentences with the same label as input and employs the T5 generation model [24] to generate augmented sentences. Liu et al. [25] proposed the SRAFBN model, which uses a self-attention mechanism to extract the key information in images so that the generated images can be of a higher quality. Liu et al. [26] used perception technology including knowledge graphs, association rules, and SNA (social network analysis), to extract features from text data, in order to solve the problem that time and operation features cannot be focused.

2.2. GAN-Based Text Generation

The SentiGAN model [7] is a commonly used text generation model; it employs multiple generators and a single discriminator to generate text for multiple classes and then uses a penalty-based objective function to generate more diverse text. The sequence generative adversarial nets (i.e., SeqGAN) model [27], which uses reinforcement learning to solve nondifferentiable problems, can be used as a text generation model. The LeakGAN model [28] allows its discriminator to leak information to its generator to solve problems pertaining to the length of generated data; this is possible because the discriminator can only provide rewards for a limited-length fragment of a sentence. Subsequently, the MANAGER method can be applied to obtain information from the discriminator, which is then passed into the generator to guide data generation. The CatGAN model [8] uses a category-aware model and hierarchical evolutionary learning algorithms to generate highly diverse samples. Cross-structure GAN (CS-GAN) [29] applies reinforcement learning to train models and includes label information to control its generator for accurate category data generation; furthermore, the CS-GAN model uses a classifier and a discriminator to support generator training. The relational generative adversarial networks (i.e., RelGAN) model [30] uses three main components to generate data. The first component is a relational memory base generator used for long-distance dependency modeling. The second component is the Gumbel Softmax distribution, which is used to solve nondifferentiable problems pertaining to discrete data. The third component is multiple embedding representation, which is implemented to help discriminators to pass informative messages into generators. The conditional Wasserstein generative adversarial network-gradient penalty (i.e., CWGAN-GP) [31] model combines a conditional GAN and a Wasserstein-GAN to increase training process stability and solve mode collapse problems.

However, the generated text method used in data augmentation is often excessively monotonic and lacking in diversity. Although GAN-based models can generate highly diverse sentences, these sentences are of poor quality. Therefore, to improve the classification of imbalanced data, the present study proposes a new framework that incorporates a hybrid EDA method, GAN-based models, and the Shapley algorithm.

2.3. Shapley Algorithm

The Shapley algorithm [9], is used for cleaning training data, and calculates the Shapley value of each training data set. The low Shapley training data points which are under a certain criterion (with the Shapley value less than or equal to zero), will be removed. The Shapley algorithm can be used for machine learning and deep learning, such as classification tasks. Classifiers such as BERT can learn from additional high-quality training data which is filtered by the Shapley algorithm. The additional training data can provide useful examples with a high semantic relationship to the original training data. In addition, Data Shapley is often used to help improve the prediction of performances [10], and originated from cooperative game theory [32]. Data Shapley calculates two players' contributions and can also be used for machine learning models to calculate the importance of the training data features. The data Shapley algorithm needs a large time complexity, so efficient calculation of the algorithm is needed [33]. In addition, the variation of the Shapley algorithm, exploits polynomial-time approximation in DNN to improve the large calculation time complexity of the Shapley value [34].

3. Methodology

The present study proposes a hybrid framework that employs generative adversarial networks and a Shapley algorithm based on easy data augmentation (HEGS) to address problems with imbalanced data in classification tasks. The EDA method can generate high-quality training sentences with a given sentence pattern by replacing words with their synonyms. Therefore, the sentences generated through EDA lack diverse sentence patterns. To address this problem, GAN-based models can be applied to learn and generate highly diverse sentence patterns based on the probability of each word in a language model. However, EDA generates high-quality sentences that exhibit low diversity, and the GAN-based model can generate sentences that are highly diverse but may deviate from semantic representations. The Shapley algorithm can calculate the similarity of multiple sentences based on the semantic space instead of grammatical structure or linguistic patterns. The present study proposes a HEGS framework that generates high-quality training sentences and improves classification performance. The HEGS framework comprises the five following stages. (1) In initial sentence generation, the EDA processing is applied to create numerous sentences to form an EDA-based augmented data set. (2) In GAN-based model training, the SentiGAN model [7] is used to learn training sentences based on the EDA-based augmented data set. An EDA-based augmented data set is used as a GAN-based model's learning objective. (3) In GAN-based model evaluation, the bilingual evaluation understudy (BLEU) and self-BLEU algorithms are used to evaluate the quality of data generated by the GAN-based model. (4) In final training sentence generation, the final augmented data set created by the GAN-based model is filtered for identical semantic sentences by the Shapley algorithm. (5) In the classification task, the BERT model is applied to train text classification tasks based on the final augmented data set and the original training data. The procedures and processes of the proposed EDA-GAN framework are presented in Figure 1.

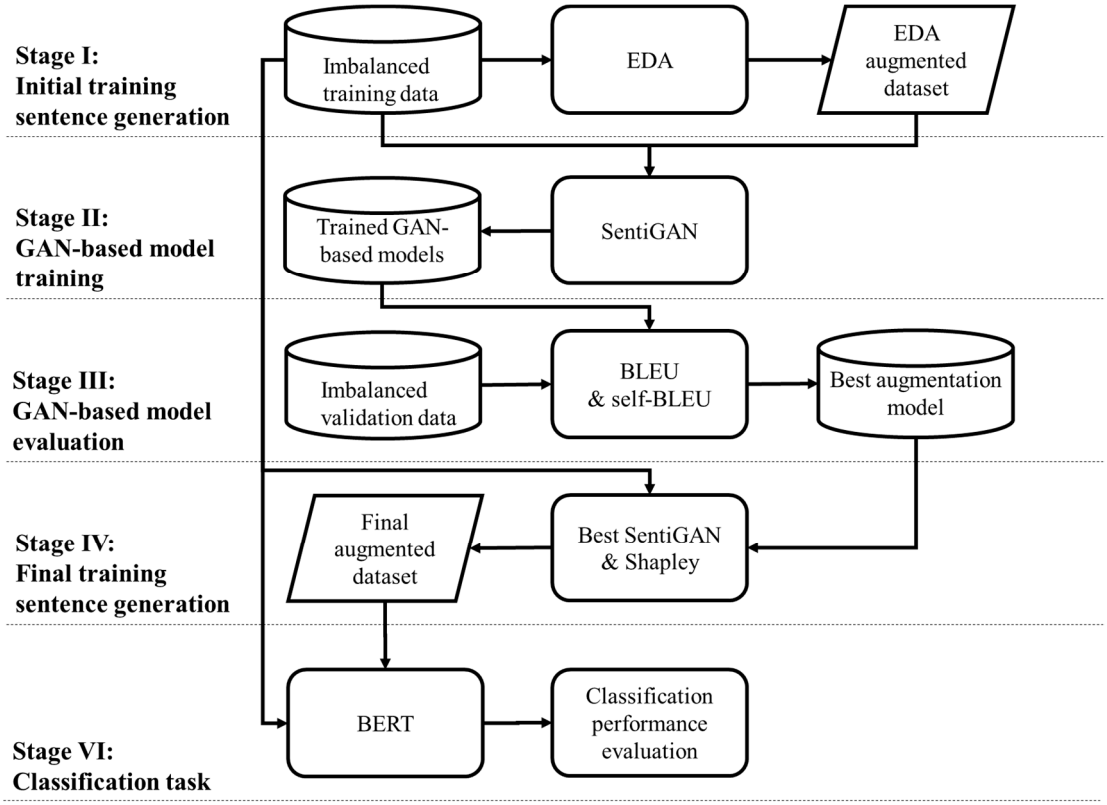


Figure 1. The flowchart of proposed HEGS framework.

3.1. Stage I: Initial Training Sentences Generated Using the EDA Method

The present study applies the EDA method to generate a set of initial training sentences. In this stage, five processes are applied to generate training data from the original training data of imbalanced data. The original training data of imbalanced data can be expressed as $D_{Train} = \{(x_r, y_r)\}_{r=1}^{r=n}$, where x_r is a word sequence and $y_r \in \{1, \ldots, C\}$ is the category that x_r falls under. The parameter α that must be set up is a ratio for determining the number of words that must be changed. For example, if the α = 0.2 and the length of a sentence is 10, then, there are two (10*0.2) words that will be changed. The imbalanced training data D_{Train} are subjected to EDA to create additional similar word sequences. The five processes are as follows:

- Synonym replacement (SR): During this process, words are randomly replaced by one of their synonyms as obtained from the WordNet library; this process is denoted as D_{SR}.
- Random insertion (RI): During this process, a number of words from Wordnet are inserted randomly; this process is denoted as D_{RI}.
- Random swapping (RS): During this process, number of words are randomly switched while retaining the same parts of speech; this process is denoted as D_{RS}.
- Random deletion (RD): Number of words are randomly deleted; this process is denoted as D_{RD}.
- Random mixing (RM): this process is expressed as $D_{RM} = \{D_{SR}, D_{RI}, D_{RS}, D_{SD}\}$, where the initial training sentences $D_{RM} = \left\{\left(\widetilde{x}_f, \widetilde{y}_f\right)\right\}_{f=1}^{\widetilde{n}}$ are generated by the previous four processes, $\widetilde{x}_f$ is the generated word sequence, $\widetilde{y}_f$ is the category that $\widetilde{x}_f$ falls under and is identical to that of the original data.

3.2. Stage II: GAN-Based Model Training Using the SentiGAN Model

In the proposed method, SentiGAN, which comprises a generator and discriminator, is used to generate highly diverse word sequences for minority classes.

3.2.1. Generator

The present study uses a modified version of the SentiGAN model (Wang and Wan, 2018). The generator of the modified SentiGAN model is built using an LSTM model, namely, $G_i(X_{t+1}\Big|S_t;\theta_g^i)$. At time step t, the generator generates the word sequence $X_{1:t} = \{X_1,\ldots,X_t\}$, and it generates the next word based on the words generated before $S_t = \{X_1,\ldots,X_t\}$. Each category in the classification task corresponds to a unique generator. At the beginning of model training, each generator is initialized by vector z, which is a vector generated from a normal distribution, and the generator then generates the next word X_t based on the previously generated word sequence S_t.

$$G_i = \begin{cases} G_i\left(z;\theta_g^i\right) & \text{initialized generator's input} \\ G_i\left(X_{t+1}\Big|S_t;\theta_g^i\right) & t \geqq 1 \end{cases} \tag{1}$$

The generated data $X_{1:t+1}$ are passed into the discriminator and are assigned the penalty probability V_{Di}^{Gi}, and the generator then updates the model parameters based on the penalty probability. The penalty-based loss function is as follows:

$$L(X) = G_i\left(X_{t+1}\Big|S_t;\theta_g^i\right)\cdot V_{Di}^{Gi}(S_t,X_{t+1}) \tag{2}$$

The total loss that the generator must minimize is as follows.

$$J_{G_i}\left(\theta_g^i\right) = \mathbb{E}_{X\sim P_{g_i}}[L(X)] = \sum_{t=0}^{t=|X|-1} G_i\left(X_{t+1}\Big|S_t;\theta_g^i\right)\cdot V_{Di}^{Gi}(S_t,X_{t+1}) \tag{3}$$

Because the discriminator can only judge a full sentence, the Monte Carlo search algorithm and roll-out policy are used to sample the final $|X|-t$ unknown tokens in incomplete sentences. Subsequently, the penalty-based loss is calculated as follows:

$$V_{Di}^{Gi}(S_{t-1},X_t) = \begin{cases} \frac{1}{N}\sum_{n=1}^{N}(1-D_i(X_{1:t}^n;\theta_d)) & t<|X| \\ 1-D_i\left(X_{1:t}^n;\theta_d\right) & t=|X| \end{cases} \tag{4}$$

where $X_{1:t}^n$ is a sample that is obtained through N-time Monte Carlo search and is based on G_i and the current state S_t. $X_{1:t}^n$ is passed into the discriminator and achieves a probability pertaining to whether the semantic similar ith category original sentences. The present study uses an LSTM model as the generator, and each generated word is determined through the Softmax function.

$$p(X_t) = softmax\left(LSTM_{\theta_g}(h_{t-1},X_{t-1})\right) \tag{5}$$

3.2.2. Discriminator

The generated sentences from the output of the generator are passed into the discriminator. The discriminator classifies both generated word sequences and original word sequences and outputs a penalty probability into the generator to update its model parameters with respect to minimal penalty loss. The discriminator is a 2-D convolutional neural network model that employs the Softmax function, and the objective function is as follows:

$$J_D(\theta_d) = -\mathbb{E}_{X\sim P_g} logD_{k+1}(X;\theta_d) - \sum_{i=1}^{k} \mathbb{E}_{X\sim P_{r_i}} logD_i(X;\theta_d) \quad (6)$$

where P_g is the distribution of generated data, $D_{k+1}(X;\theta_d)$ is the probability that the discriminator predicts that the input data are fake. P_{r_i} is the distribution of real data, r_i is the ith class, $D_i(X;\theta_d)$ is the probability that the discriminator predicts a word sequence belonging to the ith class.

3.2.3. Training Procedure

The training procedure implemented in the present study is based on the original SentiGAN model. The training data $X = \{D_{Train}, D_{RM}\}$ in the SentiGAN model comprise original training sentences and initial training sentences that are combined through EDA. First, the generator and the discriminator must train, respectively. Second, adversarial training is conducted for the generator and discriminator of the SentiGAN model. Third, the EDA-augmented data set is used as the training data for the SentiGAN model. Figure 2 presents the concept of the HEGS framework, and the detailed training procedures are as follows. The following Step one to Step five show the details of the HEGS framework, and the concept of the training procedure presented in Figure 2.

- Step one: The generator $\{G_i\}_{i=1}^{i=k}$ and discriminator $D_i(X;\theta_d)$ $i \in \{1,\ldots,k+1\}$, are randomly initialized. The training data X with a random vector by normal distribution z are used to pretrain the generator; this pretraining is based on maximum likelihood estimation.
- Step two: The generator generates the fake data, including C classes of fake data, and it uses the fake data and training data X to pretrain the discriminator $D(X;\theta_d)$.
- Step three: Every g-step performed in the generator generates sentences for C classes. First, a noise vector z is created from a normal distribution and passed into the generator $G_i(z;\theta_g^i)$. Second, the loss V_{Di}^{Gi} is obtained from the discriminator $D(X;\theta_d)$ to minimize the total loss $J_{G_i}\left(\theta_g^i\right)$.
- Step four: Every d-step performed in the discriminator updates the discriminator's model parameters through the generated sentences, which is generated from the $G_i(X\left|S;\theta_g^i\right.)\}_{i=1}^{i=k}$, and training data X, and which include original training sentences and initial training sentences obtained through EDA.
- Step five: Steps two to four are repeated until model convergence is achieved, and the final generated training sentences D_{Train_GAN} are output.

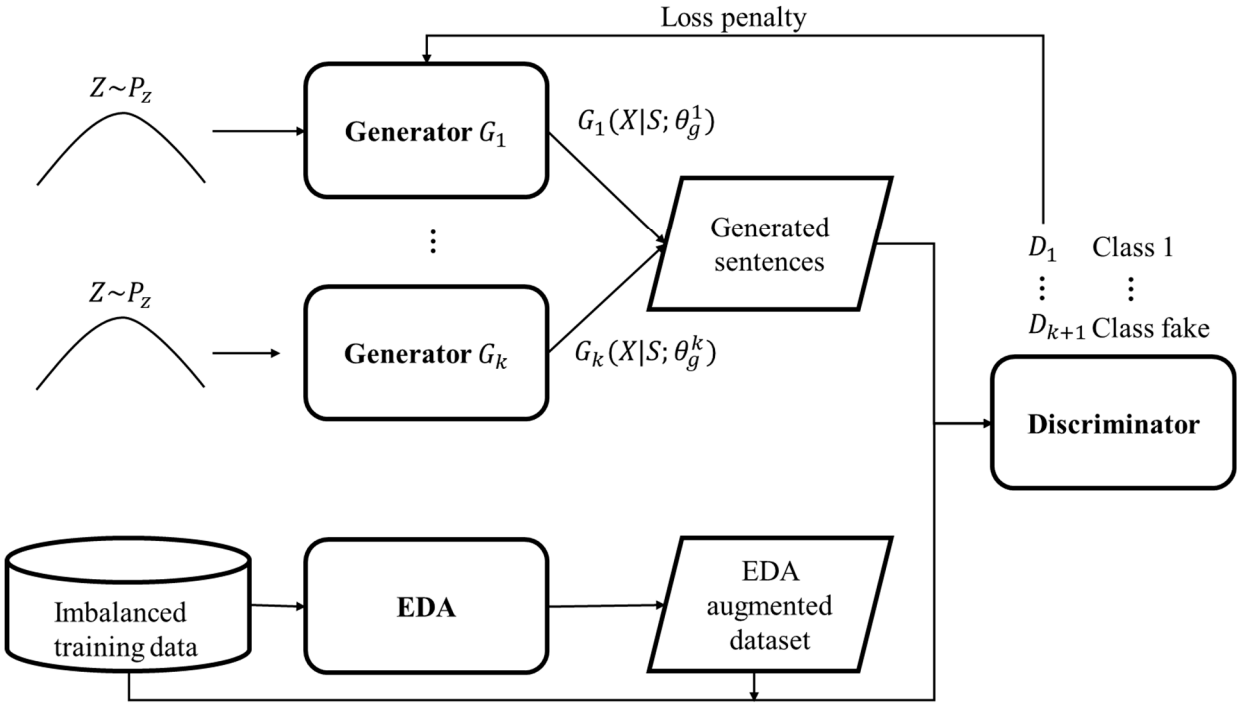


Figure 2. The model diagram of the HEGS training procedure.

3.3. Stage III: GAN-Based Model Evaluation

The present study applies BLEU [35] and self-BLEU to evaluate the sentence generation performance of the SentiGAN model. BLEU calculates the similarity of generated sentences and original sentences (original training data) and considers 2-gram measurement. In contrast, self-BLEU calculates the similarity of generated sentences independently while applying the calculation method applied for BLEU. The optimal model is identified as the model that obtains the highest average of the 2-gram BLEU (BLEU-2) and 2-gram self-gram (self-BLEU-2) in the validation set.

3.4. Stage IV: Final Training Sentence Generation by Optimal SentiGAN and Shapley Algorithm

The trained generator of the SentiGAN model is used to obtain the additional training sentences D_{Train_GAN} for minority classes. To obtain high-quality generated training sentences for the classification model, the Shapley algorithm is used to filter out reliability-generated sentences from D_{Train_GAN} to form the final augmented data set $D_{Train_Shapley}$. The processes involved in the Shapley method are as follows:

- Step one: The generated data (D_{Train_GAN}) are duplicated once, and their label is flipped. Suppose the data points are one which mean negative class, and the Shapley algorithm needs to flip the original data points to positive class.
- Step two: Validation and generated data are encoded through BERT, and the feature pertaining to the special token ([*CLS*]) is selected and used to calculate Euclidean distance.
- Step three: The Euclidean distance between each piece of generated data and validation data is calculated.
- Step four: The Euclidean distance within each piece of generated data are summed so that the distance of each generated piece of data is represented by only one number.
- Step five: Distances are sorted from minimum to maximum.
- Step six: The Shapley value is calculated by using the generated sentences with the greatest Euclidean distance. If the generated data and validation data share the same label, the Shapley value is one; otherwise, it is zero. Furthermore, the N must be considered, which is the amount of data generated. Thus, the Shapley value must divided by N.

- Step seven: To calculate the second largest piece of generated data, we must consider the value of the previous piece of generated data and check if its label is identical to that of the validation data. We must also check if the label of the previous data is identical to that of the validation data. If their labels are identical, the Shapley value is one; otherwise, it is zero, and the first number should be subtracted from the second number. Subsequently, the Shapley value is divided by Q, multiplied by the smaller number of Q or the value of previous data points, and divided by the position of the previous data. The paper sets the Q is 10.
- Step eight: The generated data undergo a removal process so that their Shapley value is negative, and they then undergo a selection process so that their label is identical to the few samples' category; this process is performed to enable the use of the data for augmentation. Finally, we combine the remaining generated data $D_{Train_Shapley}$ with the original training data D_{Train} to form the final training sentences D_{Train_Final} for the proposed framework for classifying imbalanced data.

3.5. Stage V: Classification Task

The present study uses the pretrained BERT model to classify imbalanced data. The final training data D_{Train_Final} (i.e., input) combine two sets of training data, namely, the original training sentences D_{Train} and the generated training sentences $D_{Train_Shapley}$ obtained through the SentiGAN model. In this situation, the number of training sentences corresponding to the majority and minority classes are identical (balanced), and the pretrained BERT model is finetuned.

4. Experiment

In this section, we will discuss the data set, the experimental design, and the results from the hyperparameter sensitivity, performances of GAN-based models, Shapley algorithm evaluation, performances of varied EDA methods, and the performance of augmentation models.

4.1. Data Set

The experiments of the present study were conducted on the internet movie database (IMDb) data set [36]. The IMDb data set contains 50,000 data points on movie reviews for sentiment analysis; the data are split into two categories, positive and negative. To overcome the problem of imbalanced data, the present paper randomly sampled 10% of the original IMDb data set; this was for efficiency because an overly large sample would substantially increase the time required to train the GAN model. In total, 5000 experimental samples were created and down-sampled for each class to the remaining 50% of the samples to form subsamples for minority classes (e.g., the negative class examined in the present study). Therefore, 3750 samples were created, of which 2500 were positive and 1250 were negative samples (i.e., the ratio of positive to negative samples was 2:1). In addition, the 3750 samples were split into training and testing sets at a training to testing ratio of 4:1. The training set was further split into training and validation sets at a training to validation ratio of 4:1. All 3750 samples were split into five cross-folds by stratified cross-validation procedure, so each fold was made by preserving the percentage of samples for positive and negative classes. The number of samples in each set is listed in Table 1.

Table 1. Data distribution in each fold of cross validation.

	Training Set	Validation Set	Test Set
Positive	1600	400	500
Negative	800	200	250
Total	2400	600	750

4.2. Experimental Design

An α value of 0.1 was set as the hyperparameter for EDA augmentation methods. The initial training sample augmented for the next GAN-based model was generated for the training sentences until the same number of major classes was reached for all categories. A K value of 10 was applied for the Shapley algorithm to identify the highest level of similarity between the generated samples and the validation set. In addition, a stratified five-fold cross-validation strategy was implemented to measure the classification robustness achieved by each tested generation method. The top layer of the BERT classifier was a linear layer with 768 dimensions, and the BERT classifier could learn from the IMDb data set.

4.3. Augmentation Models

The present study tested seven augmentation models for generating additional training sentences. The methods are as follows:

- EDA: This model applies EDA to generate final training sentences.
- CatGAN: This model preserves the quality and diversity of generated samples by using the hierarchical evolutionary algorithm for training and by designing evaluation metrics to filter generators.
- SentiGAN: This model incorporates multiclass generators and conducts training through reinforcement learning.
- EDA + CatGAN: This model applies EDA to generate initial training sentences and uses the CatGAN model to generate final training sentences.
- EDA + SentiGAN: This model applies the EDA method to generate initial training sentences and uses the SentiGAN model to generate final training sentences.
- EDA + CatGAN + Shapley: This model is similar to the subsequent proposed model but uses a CatGAN instead of a SentiGAN.
- EDA + SentiGAN + Shapley: This model is the proposed model of the present study.

4.4. Results on Sensitivity Analysis of the EDA Method

The experiment in the following Table 2 presents the result of sensitivity analysis for the hyperparameters α in the EDA + SentiGAN model. There are three different ratios: 0.1, 0.2, and 0.3. If the changing ratio $\alpha = 0.3$ and $\alpha = 0.5$ for the negative class, the macro self-BLEU-2 is 0.9622 and 0.9504, respectively, which is higher than $\alpha = 0.1$. A high value of self-BLEU-2 indicates a high similarity of these generated sentences by SentiGAN. On the other hand, the Macro BLEU-2 must be higher because there is high semantic similarity between the generated sentences and the original training sentences. Therefore, the $\alpha = 0.1$ is the best hyperparameter which means that these generated sentences are more diverse and can provide the BERT classifier to learn more diverse training sentences.

Table 2. Model tuning results on EDA hyperparameters of the GAN-based model.

Model	Parameter	Positive		Negative	
	α	Macro BLEU-2	Macro Self-BLEU-2	Macro BLEU-2	Macro Self-BLEU-2
EDA + SentiGAN	0.1	0.7398	0.9532	0.6340	0.9420
EDA + SentiGAN	0.3	0.7310	0.9664	0.6728	0.9622
EDA + SentiGAN	0.5	0.6580	0.9554	0.6492	0.9504

4.5. Results of the GAN-Based Models

Table 3 presents the results as obtained when macro BLEU-2 and macro self-BLEU-2 were used to measure the similarity of the sentences generated by the GAN-based models. The EDA + CatGAN model achieved a macro self-BLEU-2 value of 0.942 under the negative class, which was lower than the macro self-BLEU-2 value of 0.982 achieved by the CatGAN model. Because EDA allowed for high-quality, low-diversity sentences to be generated to form the training data for GAN-based models, the CatGAN model could generate highly diverse sentences. The result of the EDA + SentiGAN model was similar to that of the EDA + CatGAN model under the negative class. Under the positive class, the EDA + CatGAN model generated sentences that were more similar to the original negative sentences and sentences that were more diverse relative to those generated by the other models.

Table 3. Result of generated sample similarity of the BLEU-2 and self-BLEU-2 in five-fold cross validations.

Model	Positive		Negative	
	Macro BLEU-2	Macro Self-BLEU-2	Macro BLEU-2	Macro Self-BLEU-2
CatGAN	0.8270	0.9806	0.7950	0.9822
SentiGAN	0.6640	0.9156	0.6000	0.9374
EDA + CatGAN	0.8418	0.9802	0.6303	0.8122
EDA + SentiGAN	0.7398	0.9532	0.6340	0.9420

4.6. Results Obtained through the Shapley Algorithm

The Shapley algorithm was applied during postprocessing to filter out sentences with similar semantics in the GAN-based models, the macro averages obtained from the five-fold cross validation of the Shapley algorithm are listed in Table 4. The EDA + CatGAN and EDA + SentiGAN models generated an average of 2339.80 sentences in five-fold cross validation. The EDA + CatGAN + Shapley model generated a macro average of 1009.8 sentences when SP < 0, and the EDA + SentiGAN + Shapley model generated 1179.4 sentences when SP < 0. Therefore, the EDA + CatGAN + Shapley model generated more similar sentences with the semantic content of their original sentences relative to the EDA + SentiGAN + Shapley model. Overall, the EDA + CatGAN + Shapley model generated more diverse sentences (approximately 13% more diverse) relative to the EDA + SentiGAN + Shapley model.

Table 4. Statistics of the results filtered by the Shapley algorithm of the two GAN-based models.

Method	GD	SP < 0	SP > 0	POS	NEG
EDA + CatGAN + Shapley	2399.80	1009.80	3789.80	2398.80	1391.00
EDA + SentiGAN + Shapley	2399.80	1179.40	3620.20	2393.20	1227.00

Note: GD represents generated sentences; SP represents the Shapley value; POS represents positive; NEG represents negative.

4.7. Results Obtained through EDA Methods

The proposed HEGS framework is based on EDA, which is used to generate initial training sentences for training the GAN-based model. Five methods can be applied to generate sentences, and their classification performance results are listed in Table 5. The EDA(RM), EDA(SR), and EDA(RD) methods achieved a macro F1 of 0.828 on the test set, and the EDA(RS) method achieved a comparatively lower macro F1 of 0.822. This result indicates that the EDA(RM) approach, based on a combination of four EDA methods, achieved a high macro F1 and low standard deviation; that is, it generated more diverse training sentences to improve the classification of imbalanced data relative to approaches based on a single EDA method.

Table 5. Classification performances of different EDA methods in five-fold cross validation.

Model	Validation Set		Test Set	
	Macro F1	SD F1	Macro F1	SD F1
EDA(RM)	0.834	0.0114	0.828	0.0084
EDA(SR)	0.830	0.0100	0.828	0.0110
EDA(RI)	0.834	0.0084	0.826	0.0055
EDA(RS)	0.830	0.0071	0.822	0.0110
EDA(RD)	0.832	0.0110	0.828	0.0084

4.8. Results for Classification Performance of Augmentation Models

The BERT model was used as the classifier for training the subsampled IMDb data set, which achieved macro F1 values of 0.841 and 0.832 on the validation and test sets, respectively, under imbalanced data conditions (Table 6). The EDA + CatGAN + Shapley and EDA + SentiGAN + Shapley models achieved the same macro F1 of 0.854 on the test set. The EDA + SentiGAN + Shapley model was the best model because its standard deviation of 0.0055 was the lowest among the tested models. By contrast, the EDA + CatGAN and EDA + SentiGAN models achieved macro F1 values of 0.83 and 0.832 on test sets; that is, they outperformed the CatGAN and SentiGAN models because EDA enables GAN-based models to generate more high-quality training sentences. Therefore, EDA enhanced the BERT classifier by enabling it to learn more training sentences. In addition, the Shapley algorithm filtered out the generated sentences that shared the highest level of semantic similarity in the GAN-based models, increasing performance in the classification of imbalanced IMDb data.

Table 6. Classification performance based on Macro F1 and F1 standard deviation (SD) in five-fold cross validation.

Augmentation Model	Validation Set		Test Set	
	Macro F1	SD of F1	Macro F1	SD of F1
Without (only using imbalanced data)	0.841	0.0134	0.832	0.0083
EDA	0.834	0.0114	0.828	0.0084
CatGAN	0.828	0.0130	0.826	0.0055
SentiGAN	0.826	0.0090	0.824	0.0090
EDA + CatGAN	0.848	0.0123	0.830	0.0071
EDA + SentiGAN	0.834	0.0114	0.832	0.0084
EDA + CatGAN + Shapley	0.860	0.0071	0.854	0.0090
EDA + SentiGAN + Shapley	0.860	0.0071	0.854	0.0055

5. Conclusions

The present study proposed a HEGS framework that resulted in increased performance for the classification of imbalanced data during an IMDb sentiment classification task. The classification performance of the proposed random-mixing EDA method was superior to those of the random replacement (original method), random insertion, random deletion, and random swapping methods. The BLEU-2 and self-BLEU-2 measurements also indicated that the GAN-based models generated more diverse training sentences. The proposed Shapley algorithm was applied during postprocessing to filter out generated sentences from the GAN-based models and achieve the augmentation of imbalanced data, resulting in a substantially increased classification performance of both the validation and test sets. The experimental results reveal that the application of data augmentation under the proposed HEGS framework increased classification performance, which was further increased by incorporating additional training sentences. In addition, the EDA method has a very fast

processing time to generate similar sentences, but GAN-based models and BERT models need more time consuming to obtain a best trained model. Therefore, computation ability is a big challenge for our proposed framework. Future research should be focused in three directions. First, the HEGS framework could be applied to augment dialogue data and address the lack of labeled dialogue data. Second, more powerful text generation methods (e.g., GPT-2 or XLNet models) could be applied to generate more diverse sentences and further reduce the problem of imbalanced data classification. Third, different imbalanced ratios could evaluate the model performances, what number of samples of minority classes could be learned and consequently, obtain better results with the HEGS framework which would mean a stable model.

Author Contributions: Conceptualization, J.-L.W. and S.H.; methodology, J.-L.W. and S.H.; software, J.-L.W. and S.H.; validation, J.-L.W. and S.H.; formal analysis, S.H.; investigation, J.-L.W.; resources, J.-L.W. and S.H.; data curation, J.-L.W. and S.H.; writing—original draft preparation, J.-L.W. and S.H.; writing—review and editing, J.-L.W.; visualization, S.H.; supervision, J.-L.W.; project administration, J.-L.W.; funding acquisition, J.-L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by the Ministry of Science and Technology, Taiwan (Grant numbers: MOST 110-2221-E-031-004, and MOST 111-2221-E-031-004-MY3).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable as no human contact or tissue were involved.

Data Availability Statement: Data of the study are available upon request to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Abdalla, H.I.; Amer, A.A. On the Integration of Similarity Measures with Machine Learning Models to Enhance Text Classification Performance. *Inf. Sci.* **2022**, *in press*. [CrossRef]
2. Li, K.; Yan, D.; Liu, Y.; Zhu, Q. A Network-based Feature Extraction Model for Imbalanced Text Data. *Expert Syst. Appl.* **2022**, *195*, 116600. [CrossRef]
3. Lu, X.; Chen, M.; Wu, J.; Chang, P. A Novel Ensemble Decision Tree Based on Under-Sampling and Clonal Selection for Web Spam Detection. *Pattern Anal. Appl.* **2018**, *21*, 741–754. [CrossRef]
4. Liu, S.; Zhang, K. Under-sampling and Feature Selection Algorithms for S2SMLP. *IEEE Access* **2020**, *8*, 191803–191814. [CrossRef]
5. Wei, J.; Zou, K. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics: Stroudsburg, PA, USA; pp. 6382–6388. [CrossRef]
6. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Bengio, Y. Generative adversarial nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2 (NIPS'14), Montreal, QC, Canada, 8–13 December 2014; MIT Press: Cambridge, MA, USA; pp. 2672–2680.
7. Wang, K.; Wan, X. SentiGAN: Generating Sentimental Texts via Mixture Adversarial Networks. In Proceedings of the IJCAI, Stockholm, Sweden, 13–19 July 2018; pp. 4446–4452. [CrossRef]
8. Liu, Z.; Wang, J.; Liang, Z. CatGAN: Category-Aware Generative Adversarial Networks with Hierarchical Evolutionary Learning for Category Text Generation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8425–8432. [CrossRef]
9. Liang, W.; Liang, K.H.; Yu, Z. HERALD: An Annotation Efficient Method to Detect User Disengagement in Social Conversations. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Bangkok, Thailand, 1–6 August 2021; Association for Computational Linguistics: Stroudsburg, PA, USA; Volume 1, pp. 3652–3665. [CrossRef]
10. Ghorbani, A.; Zou, J. Data Shapley: Equitable Valuation of Data for Machine Learning. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 2242–2251.
11. Wu, J.; Chung, W. Sentiment-based masked language modeling for improving sentence-level valence–arousal prediction. *Appl. Intell.* **2022**, *in press*. [CrossRef]

12. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2018; Volume 1, pp. 4171–4186. [CrossRef]
13. Jin, D.; Jin, Z.; Zhou, J.T.; Szolovits, P. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8018–8025. [CrossRef]
14. Garg, S.; Ramakrishnan, G. Bae: Bert-Based Adversarial Examples for Text Classification. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 8–12 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA; pp. 6174–6181. [CrossRef]
15. Zhao, M.; Zhang, L.; Xu, Y.; Ding, J.; Guan, J.; Zhou, S. EPiDA: An Easy Plug-in Data Augmentation Framework for High Performance Text Classification. *arXiv* **2022**, arXiv:2204.11205.
16. Karimi, A.; Rossi, L.; Prati, A. AEDA: An Easier Data Augmentation Technique for Text Classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 2748–2754. [CrossRef]
17. Ren, S.; Zhang, J.; Li, L.; Sun, X.; Zhou, J. Text AutoAugment: Learning Compositional Augmentation Policy for Text Classification. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; Association for Computational Linguistics: Stroudsburg, PA, USA; pp. 9029–9043. [CrossRef]
18. Kobayashi, S. Contextual augmentation: Data Augmentation by Words with PARADIGMATIC relations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; Association for Computational Linguistics: Stroudsburg, PA, USA; Volume 2, pp. 452–457. [CrossRef]
19. Wu, X.; Lv, S.; Zang, L.; Han, J.; Hu, S. Conditional Bert Contextual Augmentation. In Proceedings of the International Conference on Computational Science, Faro, Portugal, 12–14 June 2019; Springer: Cham, Switzerland; pp. 84–95. [CrossRef]
20. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
21. Anaby-Tavor, A.; Carmeli, B.; Goldbraich, E.; Kantor, A.; Kour, G.; Shlomov, S.; Tepper, N.; Zwerdling, N. Do Not Have Enough Data? Deep Learning to the Rescue! In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 7383–7390. [CrossRef]
22. Wu, X.; Gao, C.; Lin, M.; Zang, L.; Wang, Z.; Hu, S. Text Smoothing: Enhance Various Data Augmentation Methods on Text Classification Tasks. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; Association for Computational Linguistics: Stroudsburg, PA, USA; Volume 2, pp. 871–875. [CrossRef]
23. Jo, B.C.; Heo, T.S.; Park, Y.; Yoo, Y.; Cho, W.I.; Kim, K. DAGAM: Data Augmentation with Generation and Modification. *arXiv* **2022**, arXiv:2204.02633.
24. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
25. Liu, X.; Chen, S.; Song, L.; Wozniak, M.; Liu, S. Self-attention Negative Feedback Network for Real-time Image Super-Resolution, Journal of King Saud University. *Comput. Inf. Sci.* **2021**, *34*, 6179–6186. [CrossRef]
26. Liu, S.; He, T.; Li, J.; Li, Y.; Kumar, A. An Effective Learning Evaluation Method Based on Text Data with Real-time Attribution—A Case Study for Mathematical Class with Students of Junior Middle School in China. *ACM Trans. Asian Low Resour. Lang. Inf. Process* **2022**, *10*, 3474367. [CrossRef]
27. Yu, L.; Zhang, W.; Wang, J.; Yu, Y. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
28. Guo, J.; Lu, S.; Cai, H.; Zhang, W.; Yu, Y.; Wang, J. Long Text Generation via Adversarial Training with Leaked Information. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32. Available online: https://ojs.aaai.org/index.php/AAAI/article/view/11957 (accessed on 1 August 2022).
29. Li, Y.; Pan, Q.; Wang, S.; Yang, T.; Cambria, E. A generative model for category text generation. *Inf. Sci.* **2018**, *450*, 301–315. [CrossRef]
30. Nie, W.; Narodytska, N.; Patel, A. Relgan: Relational Generative Adversarial Networks for Text Generation. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
31. Zheng, M.; Li, T.; Zhu, R.; Tang, Y.; Tang, M.; Lin, L.; Ma, Z. Conditional Wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification. *Inf. Sci.* **2020**, *512*, 1009–1023. [CrossRef]
32. Kumar, I.E.; Venkatasubramanian, S.; Scheidegger, C.; Friedler, S. Problems with Shapley-value-based explanations as feature importance measures. In Proceedings of the International Conference on Machine Learning, Virtual Event, 24–26 November 2020; pp. 5491–5500. [CrossRef]
33. Jia, R.; Dao, D.; Wang, B.; Hubis, F.A.; Hynes, N.; Gürel, N.M.; Spanos, C.J. Towards Efficient Data Valuation Based on the Shapley Value. In Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, Okinawa, Japan, 16 April 2019; pp. 1167–1176. [CrossRef]

34. Ancona, M.; Oztireli, C.; Gross, M. Explaining Deep Neural Networks with A Polynomial Time Algorithm for Shap-LEY value Approximation. In Proceedings of the 2019 International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 272–281. [CrossRef]
35. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, Philadephia, PA, USA, 6–12 July 2002; pp. 311–318. [CrossRef]
36. Maas, A.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 142–150.