TAMS: Text Augmentation using Most Similar Synonyms for SMS Spam Filtering

Mohammad Qussai Jouban, Zakarya Farou

ELTE Eötvös Loránd University, Department of Data Science and Engineering, Institute of Industry - Academia Innovation, Budapest, Hungary

Abstract

Spam filtering is a non-standard derivative data science problem aiming to catch unsolicited and undesirable messages and prevent those messages from reaching a user's inbox. To solve the abovementioned problem, we propose a text augmentation approach using the most similar synonyms called TAMS. We used Random forest and Bidirectional LSTM classification models for the experimental part to assess the proposed approach. The results indicate that training the classifiers with synthesized spam messages generated by TAMS reduces the influence of the imbalance problem present by nature in the dataset and improves the overall performance of the classification models. Hence, this study shows the potential of using TAMS to enhance the classification performance on textual data where the imbalance scenario is present.

Keywords

Spam filtering, Text classification, Text Augmentation, Imbalance learning

1. Introduction

Spam filtering is a non-standard derivative data science problem [1]. Derivative because it is an extension of core problems, i.e., classification problems. Non-standard, since the data has an unusual distribution on the target variable, such problems belong to the imbalance problems family. Spam filtering is also one of the most common problems in the Natural Language Processing domain. The main target of this problem is to identify the spam messages and filter them out from the legitimate messages. Solving this problem will be very beneficial for telecommunication companies, where text messaging is the most common non-voice use of a mobile phone. In fact, according to security firm Cloudmark, about 30 million spam messages are sent to cell phone users across North America, Europe, and the U.K.

This study aims to improve the SMS spam filtering by solving the most common problem in the available datasets, which is the imbalance problem, where most of the samples belong to the legitimate class, i.e., legitimate messages, the so-called majority class C^-, and a small proportion of the samples belongs to the spam class, i.e., spam messages, so-called minority class C^+. Dealing with an imbalanced dataset is one of the main challenges in machine learning, especially in classification problems, where most well-known classification models tend to be biased toward the majority class and fail to identify the minority class. The most common solution to the imbalanced datasets problem is generating new samples belonging to the minority class to make the dataset balanced.

The paper is organized as follows: Section 2 describes the imbalance problem, data augmentation, used machine learning models, some related terminologies and related works. Section 3 introduces the proposed text augmentation approach TAMS with a detailed explanation and practical examples. The experimental results, including the dataset, evaluation metrics, and results with discussion, are presented in Section 4. Lastly, Section 5 outlines the conclusion about the conducted research and the potential research direction to improve learning and classification of similar problems.

2. Background

Data science techniques are used to solve many problems. These methods can learn and likely extract hidden patterns from the data used as input. Regardless of data modality (e.g., textual, visual, tabular), we classify data science problems into standard and non-standard problems. Standards problems mainly concern supervised learning (predictive problems) and unsupervised learning (descriptive problems). However, there exist more complex (non-standard) problems than the cited ones. These complex problems are derived or hybridized from the standard, i.e., core problems.

2.1. Imbalance problem

Imbalance problem occurs when the target class has very few samples opposed to the other classes, It is classified as a non-standard derivative problem.

ITAT'22: Information technologies – Applications and Theory, September 23–27, 2022, Zuberec, Slovakia

*Zakarya Farou.

f7rg3j@@inf.elte.hu (M. Q. Jouban); zakaryafarou@inf.elte.hu (Z. Farou)

0000-0003-3996-2656 (Z. Farou)

Technically, we call a dataset imbalanced regardless of its data modality when there is a disproportion among the number of instances of each class, making classes under-represented. Therefore, traditional machine learning (ML) algorithms have complications defining the target class's decision boundaries.

As various real-world applications and diverse domains fall under the imbalance problem, researches on imbalanced data classification have expanded and gained more interest [2]. Mainly, we face the imbalance problem in credit card fraud detection [3], anomaly detection [4], e-mail foldering [5], medical diagnosis [6] [7] [8], particles identification [9], face recognition [10], fault diagnosis [11] [12], text classification [13] [14], and many others.

2.2. E-Mail Spam Filtering

Spam emails, also known as junk emails, are messages transmitted by spammers via email. Users are confronting several issues such as the abuse of traffic, limited storage space, computational power, waste of users' time, and threat to user security. Therefore, appropriate email filtering is essential to provide more security and increase the efficacy of end users. Data Scientists conducted several types of research on email filtering; some achieved good accuracy, and some continued. For instance, in [15], the authors developed a mobile SMS spam filtering for Nepali text and used Naïve bayesian and support vector machines as classifiers, while in [16], Mohammed et al. present an approach for filtering spam email using machine learning algorithms. At first, they used the tokenization method to filter spam and ham words from the training data and utilized them to create testing and training tables and experienced with various data mining algorithms. Furthermore, Singh et al. [17] discussed the solution and classification process of spam filtering and presented a combining classification technique to get better spam filtering results. Other studies such as [18] proposed a method for detecting malicious spam through feature selection and improving the training time and accuracy of a malicious spam detection system.

Despite the numerous proposals, most anti-spam strategies have some inconsistency between false negatives (missed spam) and false positives (rejecting good emails) due to imbalance problems that act as an obstacle for most systems to make anti-spam systems successful. Therefore, an adequate spam-filtering system that addresses imbalance issues is the prime demand for web users. Recently, the authors in [19] presented an improved random forest for text classification that incorporates bootstrapping and random subspace methods simultaneously and tested its performance on the SMS binary class dataset. The method removes inessential features, adds some trees in the forest on each iteration, and monitors the classification performance of RF.

2.3. Data augmentation for textual data

Obtaining accurate results while training a classifier becomes difficult due to the lack of available, varied, and meaningful data, especially when the imbalance problem is present. Therefore, additional samples should be added to train the classifier more efficiently. However, gathering such data is time-consuming and needs domain experts that examine and annotate the data.

As assembling such data is costly, synthesizing new data from the existing ones seems to be a promising approach, specifically if the quality of the generated data is as good as the original one. In the data science ecosystem, increasing the training dataset, i.e., generating additional samples from the existing ones, is known as data augmentation. For an imbalanced class problem, data augmentation would help avoid overfitting, reduce the bias of the classifiers toward the majority class, and improving the generalization ability of the trained models. However, text augmentation would only be worthwhile if the generated data has new linguistic patterns that are pertinent to the task and have not yet been seen in pre-training.

In NLP, there are numerous data augmentation techniques, such as paraphrasing [20], close embeddings [21], swapping [22], inducing spelling mistakes [23], deleting [24], and synonyms replacement [25][26][27].

For this study, we mainly focus on local data augmentation, particularly token substitution, because it is a cost-effective and easily accessible yet powerful textual data augmentation method. Token substitution is a popular method that replaces a token in the sentence with its synonym.

2.4. Supervised machine learning

Machine learning [28] is a form of artificial intelligence that enables a system to learn from data rather than through explicit programming. We can use supervised learning algorithms for non-standard derivative problems such as imbalance learning, as both data and its desired label are present. In this paper, we are considering only two classifiers, random forest and bidirectional LSTM.

2.4.1. Random forests

according to [29], random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. Random forests have similar hyperparameters to the Decision Tree. In addition, they have a very important hyper-

parameter, which is the number of estimators, i.e., the number of trees in the forest.

2.4.2. Bidirectional LSTM

Hochreiter and Schmidhuber firstly proposed LSTM back in 1997 to overcome the gradient vanishing problem of RNN [30]. Its main idea is to introduce an adaptive gating mechanism, which decides the degree to keep the previous state and memorize the extracted features of the current data input. LSTM models can recognize the relationship between values at the beginning and end of a sequence. For the sequence modeling tasks, it is beneficial to have access to the past and future contexts. By the end of 1997, Schussed and Palatal proposed BiLSTM to extend the unidirectional LSTM by introducing a second hidden layer, where the hidden to hidden connections flow in the opposite temporal order. Therefore, the model can exploit information from both the past and the future, which can improve model performance on sequence classification problems. BiLSTM models are primarily used in natural language processing applications like text classification because BiLSTM is a powerful tool for modeling the sequential dependencies between the words and phrases in both directions of the sequence.

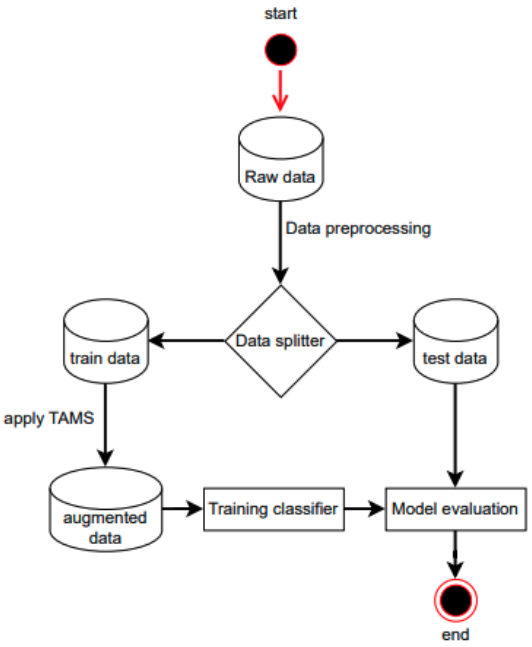


Figure 1: Data augmentation based on the proposed TAMS

3. Text augmentation using most similar synonyms

The diagram displayed in Fig. 1 summarizes the integration between the proposed text augmentation approach TAMS and the standard supervised learning training and evaluation process. It starts with data cleaning and common NLP preprocessing steps. After that, the training set is augmented by using TAMS. TAMS generates synonyms for each word, then filters them, and keeps the most similar synonyms to generate new messages. Then the augmented data will be used to train the chosen classifiers defined in Section 2.4 and evaluate their performance on the test set.

3.1. Data cleaning and preprocessing

To prepare the textual data for the model building we performed the following text preprocessing steps:

- **Duplicates removal:** duplicate samples are problematic as when the same sample appears more than once; it receives a disproportionate weight during the training phase. Thus models that succeed in recurring instances will look like they perform well, while in reality, this is not the case. Additionally, duplicate samples can ruin the split between train, validation, and test sets in cases where identical entries are not all in the same set, leading to biased performance estimates leading to disappointing models in the prediction phase.
- **Tokenization:** this step aims to split each message into a list of words, and this is necessary for two reasons, it is required to recognize the stop word and remove them in the next step, and it is also a requirement to use the Word2Vec to compute a continuous vector representation for each word in the message.
- **Removal of stop words:** stop words are the most frequent words in any language, such as articles, prepositions, pronouns, and conjunctions. They do not add much information to the text. Examples of stop words in English are words like the, a, an, so, what, and many more. Stop words are available in abundance in any human language. By removing these words, we remove the low-level information from the text to focus on critical ones.
- **Message representation:** this step aims to compute a numerical representation for each message by computing a vector that represents the message simply by taking the average of vectors representing each word in that message, where these vectors are computed using the Word2Vec model. The resulting vector will be the feature vector of the message. Word2Vec [31] is a model that computes continuous vector representations of words

from large data sets. These word representations help establish the relationship between a word and the other similar meaning words through the created vector representations. Word2Vec models produce real-valued vectors, which allow the machine learning algorithm to deal with the textual data, and at the same time, these vectors keep the semantic meaning of the represented words, where the similar meaning words are closer in space, which indicates their semantic similarity.

3.2. Proposed text augmentation method

The proposed text augmentation method aims to generate new spam messages based on the original ones by replacing some words in the message with their most similar synonyms. Fig. 2 summarizes the proposed TAMS approach, starting from the preprocessed message text tokens and it ends by generating a set of semantically similar messages. In the following subsections, we will explain each step in detail.

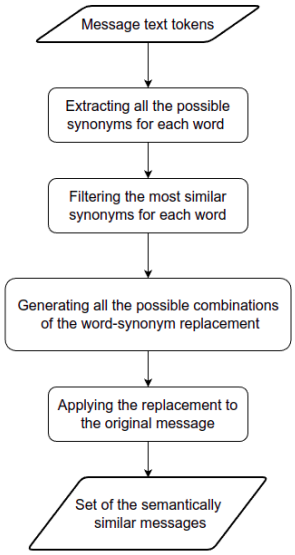


Figure 2: Summary of TAMS approach.

- **Synonyms extraction:** extracting all possible synonyms for each word in the sentence is done with the help of the WordNet database. WordNet is a lexical database of semantic relations between words introduced by [32]. It links words into semantic relations, including synonyms, antonyms, hyponyms, and other morphological relations.
  Fig. 3 shows synonyms of the word *Car* in a tree-like structure where the tree's root is the main word, and each node of the first level of the tree represents synonyms. Each synonym is connected to the root via an edge with a weight representing its similarity.

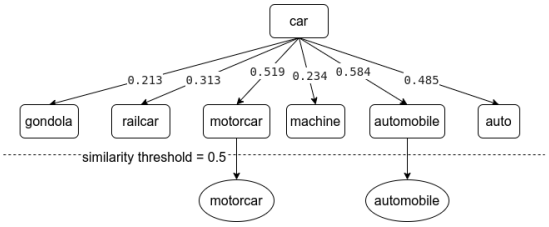


Figure 3: Synonyms extraction and filtering process

- **Finding most similar synonyms:** In order to choose the most similar synonyms, every synonym is represented using the Word2Vec representation, and the cosine distance is used to measure the similarity between the word and its synonyms.
  The most similar synonyms are the synonyms that have a similitude greater than or equal to a predefined similarity threshold S_T. Fig. 3 shows the most similar synonyms for the word *car* in case $S_T = 0.5$.
  Optimizing S_T is essential as it explicitly impacts the classification performances. A grid search-like process is done to discover the optimal S_T by training multiple models using diverse augmented data according to candidate similarity thresholds. Candidate similarity thresholds are $S_T = [0.625, 0.65, 0.675, 0.7, 0.725, 0.75]$. These candidates are selected according to the augmented data's spam percentage S_P. For example, by using $S_T = 0.625$, TAMS will generate data with an $S_P = 55.98\%$, in this case, the augmented data is approximately balanced. However, lower values ($S_T < 0.625$) will yield an imbalanced data situation. For the highest candidate value i.e., $S_T = 0.75$, the corresponding spam percentage is 20.10%, and for higher values($S_T > 0.75$), the augmented data will be the same as the original data with a high imbalance. To determine the best threshold S_T, We calculate a rank R for each candidate as shown in Eq 1:

$$R = \frac{r(SC) + r(BH) + r(MCC) + r(F_1)}{4} \tag{1}$$

R is the mean average of Spam Caught (SC), Blocked Ham (BH), Matthews Correlation Coefficient (MCC), and F1 Score (F_1) ranks divided

Table 1
Random forest and Bidirectional LSTM models with similarity threshold grid search results.

classification model	S_T	S_P	MCC	SC	BH	F_1	R
Random forest	0.625	55.98	**0.8602**	**0.8321**	0.99	**0.8755**	**5**
	0.65	46.43	0.8358	0.7939	0.99	0.8525	3
	0.675	40.63	0.8496	0.8015	0.77	0.8642	4.25
	0.7	32.09	0.8539	0.7939	0.55	0.8667	4.75
	0.725	24.12	0.8295	0.7481	**0.44**	0.8412	3
	0.75	20.10	0.7998	0.7023	**0.44**	0.8105	2.25
Bidirectional LSTM	0.625	55.98	0.9296	**0.9313**	0.77	0.9384	3.5
	0.65	46.43	**0.9336**	0.9237	0.55	**0.9416**	**5.25**
	0.675	40.63	0.9332	0.9008	0.22	0.9402	4.5
	0.7	32.09	0.9196	0.8626	**0.0**	0.9262	2.25
	0.725	24.12	0.9286	0.8931	0.22	0.936	2.75
	0.75	20.10	0.9293	0.9237	0.66	0.938	3.5

by the number of metrics (these metrics are defined in Section 4.3). We rank the scores in descending order for each metric between 6 and 1 (6 is the number of candidate similarity thresholds). We give 6 for the best metric score for a specific S_T and 1 for the worst metric score.
Table 1 show the result of six random forest models and six Bidirectional LSTM models. Each model was trained using a different augmented training set according to the candidate similarity thresholds specified above. The best R for each model is used in the experimental part.

- **Text augmentation:** After specifying the most similar synonyms for each word in a given message, text augmentation is done by generating all the possible combinations of the word-synonym replacement and applying the replacement to the original message, where each replacement generates a new message semantically similar to the original message. Table 2 shows an example of the TAMS approach with a threshold $S_T = 0.6$, where three new messages are generated.

Table 2
Example of text augmentation with a message.

Original text	Defer admission till next year
Augmented text	**Postpone** admission till next year Defer **admittance** till next year **Postpone admittance** till next year

- **Expending the training set size:** Once the text augmentation is done, we append the generated textual data with the original training set and use the extended training set to train the classifiers.

4. Experiments and results

The code for the experimental part is done using Google Colab environment and is available via this link [1]

4.1. Dataset description

We used the SMS Spam Collection dataset [33] for the experimental part. The dataset has 5574 SMS messages distributed as the following: 4825 Ham messages with a percentage of 86.6 and 747 Spam messages with a percentage of 13.4, which means that the SMS Spam Collection dataset has an imbalanced ratio of $IR = 6.46$. IR [34] for binary classification problems is computed by Eq 2.

$$IR = \frac{C^{-}_{size}}{C^{+}_{size}}, where\ IR \geq 1 \tag{2}$$

where, C^{-}_{size} and C^{+}_{size} represents majority and minority class sizes respectively.

4.2. Train test split

In order to to make the evaluation process accurate and realistic, the dataset was split in a stratified way, i.e., the spam messages percentage S_p and the ham messages parentage H_p are approximately the same in both the training set and the testing set. The train set and test set statistics are shown in Table 3. The test set contains 1034 messages with a percentage of 20% from the original dataset, while the train set contains 4135 messages.

4.3. Evaluation Metrics

Confusion matrix, Matthews Correlation Coefficient, Spam Caught, Blocked Hams, and F1 score are used to

[1] https://colab.research.google.com/drive/12LGx9j6OtEIadERJXJtmqaBaUaqVjz9S?usp=sharing

Table 3
The train set and test set statistics.

Set	S_p %	H_p %	number of SMS
Training	12.6 %	87.4 %	4135
Testing	12.7 %	87.3 %	1034

evaluate and compare the proposed text augmentation method and measure the performance of SMS spam filters.

Table 4
Confusion Matrix.

	$\mathbf{C}^+$	$\mathbf{C}^-$
$\mathbf{C}^+$	True Positives (TP)	False Negatives (FN)
$\mathbf{C}^-$	False Positives (FP)	True Negatives (TN)

4.3.1. Confusion Matrix

is a technique for summarizing the prediction results of a classification model, Table. 4 well defines the Confusion Matrix (CM) for binary classification problems.

4.3.2. F1 score

is Precision-Recall trade-Off i.e. it combines the precision and recall metrics into a single metric, and it is calculated using Eq. 3, F1 score has been designed to work well on imbalanced data.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

Where $Precision$ is computed by Eq 4:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

While the $Recall$ is computed by Eq 5

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

4.3.3. Matthews Correlation Coefficient

is used to measure to the quality of the binary classifications, and this measure takes values in the range [-1,+1], where +1 means a perfect predication, 0 indicates the random prediction and -1 means an inverse prediction, and it is given by Eq. 6

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{W}} \tag{6}$$

Where: $W = (TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)$.

4.3.4. Spam Caught

is equivalent to the True Positive Rate (TPR) or Recall, and it means the number of the spam messages which are detected by the spam filter over the number of all the spam messages,i.e. it is the measure of correctly identifying True Positives by the model, and it is defined by the Eq. 7

$$SC = \frac{TP}{TP + FN} \tag{7}$$

4.3.5. Blocked Hams

is equivalent to the False Positive Rate (FPR). A low score close to 0 is preferred as it reflects that we have few false predictions. Furthermore, FPR is the number of legitimate messages which are classified as spam by the spam filter over the number of all the legitimate messages, and it is defined in Eq. 8

$$BH = \frac{FP}{FP + TN} \tag{8}$$

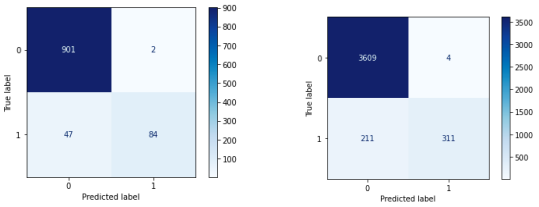
4.4. Experimental results

4.4.1. Random Forest

In this experiment, a random forest classifier with its hyperparameters: $n\_estimators = 200$, $min\_samples\_split = 20$, $max\_features = 25$, and $Gini$ criterion is used to implement the spam filter.

The first part of the experiment depends only on the original data i.e. the imbalanced data, Fig. 4a shows the confusion matrix which summaries the trained model's predictions on the test set, and Fig. 4b shows the results of the 10-folds cross-validation (10-CV) applied on the original training set, while Table 5 shows the resulting evaluation measures based on the test set.

Figure 4: Confusion matrices of RF trained on original training data.



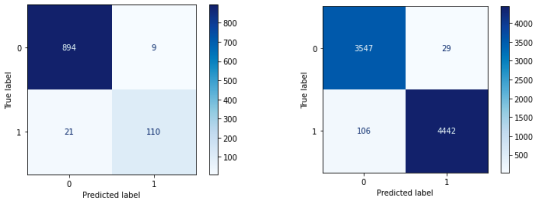
(a) confusion matrix of RF model tested on the test data.

(b) CM of RF model with 10-CV applied on the training data.

In the second part of this experiment, a random forest model with the same hyperparameters is trained using the augmented data based on the proposed TAMS.

As previously discussed, the optimal threshold S_T is determined based on Table 1. Therefore, we choose $S_T = 0.625$ as it has the highest rank R. Results of TAMS-RF are displayed in Fig. 5a, which shows the confusion matrix obtained using the test set solely, and Fig. 5b shows the results of the 10-folds cross-validation applied on the augmented training set.

Figure 5: Confusion matrices of TAMS-RF



(a) confusion matrix of TAMS-RF model tested on the test data.

(b) CM of TAMS-RF model with 10-CV applied on the training data.

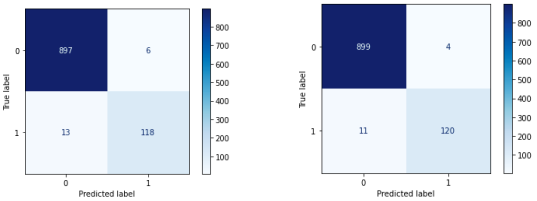
As Fig 4, Fig 5, and Table 5 shows, the model trained on the augmented data using the TAMS approach (TAMS-RF) performs better than the model trained on the original data solely in most used evaluation metrics. TAMS improved the MCC by 12.36%, SC by 30.96%, and F_1-score by 13.67%, which means that the proposed text augmentation improved spam detection meaning that we reduced the bias of RF toward the Ham class and improved the generalization ability of the models. We can conclude that TAMS generated data that has new linguistic patterns that are pertinent to the task and have not yet been seen in pre-training. However, the trained RF with the original data has a better BH score than TAMS-RF, but that does not mean it is better than the second one. On the contrary, it means that the first model is biased toward the ham class and has fewer spam predictions, i.e.; it could not detect the spam messages properly.

4.4.2. Bidirectional LSTM

In this experiment, a Bidirectional LSTM model with $Adam$ optimizer and $Sigmoid$ activation function at the output layer is used to implement the spam filter. It was trained with a Batch size of 10 for ten epochs. f

Similarly to RF, the first part of the experiment depends only on the original data, Fig. 6a shows the confusion matrix with $FP = 6$ and $FN = 13$, and Table 5 shows the resulting values of evaluation metrics. While in the second part, a BiLSTM model with the same structure is trained using the augmented data generated by TAMS. For TAMS-BiLSTM, we used $S_T = 0.65$ as it has the highest rank R (see Table 1).

Figure 6: Confusion matrices of two BiLSTM models, the first model is trained using the original data, and the second model using the augmented data.



(a) CM of BiLSTM model trained using the none-augmented dataset.

(b) CM of BiLSTM model trained by augmented data according to TAMS

The confusion matrix in Fig. 6b shows a decrease in false predictions as $FP = 4$, and $FN = 11$. While Table 5 shows that the TAMS-BiLSTM model overcomes BiLSTM according to all the suggested evaluation metrics, and the improvements are as follows: MCC by 2%, Sc by 1.7 %, BH by 33%, F1-score by 1.76%.

Both experiments proved that augmenting the training set by using the proposed TAMS approach helped the classification models to increase their ability to detect spam messages and not get biased toward the majority class and that synthesizing new data from the existing ones is indeed an excellent alternative to data collection and annotation.

5. Conclusion

Nowadays, the spam filtering task is still a real challenge because most of the available datasets are imbalanced. Dealing with such non-standard derivative datasets is a common problem in classification tasks, especially in the spam filtering case. We proposed TAMS, a text augmentation based on the most similar synonyms replacement to enhance the quality of supervised learning models and solve the spam filtering problem. Experimental results showed that generating additional samples from the existing ones using TAMS added new linguistic patterns pertinent to the task and helped in improving the classification performance of traditional classifiers like the random forest and deep learning models like the Bidirectional LSTM. We can deduce that TAMS increased the ability of both used models to identify spam messages, reduce the bias toward the majority class, and improve the trained models' generalization ability.

In future work, we aim to improve TAMS further and enhance the quality of its generated data. Furthermore, we have to test our method on other textual datasets and compare it with other text augmentation methods to ensure that the proposed model is generic and not

Table 5
Summary of the experimental results.

Classification model	Train set	MCC	SC	BH	F_1
Random forest	OD	0.7697	0.6412	**0.22**	0.7742
	TAMS	**0.8648**	**0.8397**	0.99	**0.88**
Bidirectional LSTM	OD	0.9155	0.9007	0.66	0.9255
	TAMS	**0.9334**	**0.916**	**0.44**	**0.9418**

specific to spam filtering exclusively.

Acknowledgments

This research is supported by the ÚNKP-21-3 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund.

References

[1] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, F. Herrera, Learning from imbalanced data sets, volume 11, Springer, 2018.

[2] F. Thabtah, S. Hammoud, F. Kamalov, A. Gonsalves, Data imbalance in classification: Experimental evaluation, Information Sciences 513 (2020) 429–441.

[3] N. Malave, A. V. Nimkar, A survey on effects of class imbalance in data pre-processing stage of classification problem, International Journal of Computational Systems Engineering 6 (2020) 63–75.

[4] Q. Chen, A. Zhang, T. Huang, Q. He, Y. Song, Imbalanced dataset-based echo state networks for anomaly detection, Neural Computing and Applications 32 (2020) 3685–3694.

[5] P. Bermejo, J. A. Gámez, J. M. Puerta, Improving the performance of naive bayes multinomial in e-mail foldering by introducing distribution-based balance of datasets, Expert Systems with Applications 38 (2011) 2072–2080.

[6] D. Gan, J. Shen, B. An, M. Xu, N. Liu, Integrating tanbn with cost sensitive classification algorithm for imbalanced data in medical diagnosis, Computers & Industrial Engineering 140 (2020) 106266.

[7] M. Kinal, M. Woźniak, Data preprocessing for desknn and its application to imbalanced medical data classification, in: Asian Conference on Intelligent Information and Database Systems, Springer, 2020, pp. 589–599.

[8] Z. Farou, N. Mouhoub, T. Horváth, Data generation using gene expression generator, in: C. Analide, P. Novais, D. Camacho, H. Yin (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2020, Springer International Publishing, Cham, 2020, pp. 54–65.

[9] Z. Farou, S. Ouaari, B. Domian, T. Horváth, Directed undersampling using active learning for particle identification, in: Recent Innovations in Computing, Springer, 2022, pp. 149–162.

[10] X. Bai, Y. Hu, P. Zhou, F. Shang, S. Shen, Data augmentation imbalance for imbalanced attribute classification, arXiv preprint arXiv:2004.13628 (2020).

[11] W. Zhang, X. Li, X.-D. Jia, H. Ma, Z. Luo, X. Li, Machinery fault diagnosis with imbalanced data using deep generative adversarial networks, Measurement 152 (2020) 107377.

[12] W. Hao, F. Liu, Imbalanced data fault diagnosis based on an evolutionary online sequential extreme learning machine, Symmetry 12 (2020) 1204.

[13] Y. Liu, H. T. Loh, A. Sun, Imbalanced text classification: A term weighting approach, Expert systems with Applications 36 (2009) 690–701.

[14] J. Jang, Y. Kim, K. Choi, S. Suh, Sequential targeting: an incremental learning approach for data imbalance in text classification, arXiv preprint arXiv:2011.10216 (2020).

[15] T. B. Shahi, A. Yadav, et al., Mobile sms spam filtering for nepali text using naïve bayesian and support vector machine, International Journal of Intelligence Science 4 (2014) 24–28.

[16] S. Mohammed, O. Mohammed, J. Fiaidhi, S. Fong, T. H. Kim, Classifying unsolicited bulk email (ube) using python machine learning techniques, International Journal of Hybrid Information Technology 6 (2013) 43–56.

[17] V. K. Singh, S. Bhardwaj, Spam mail detection using classification techniques and global training set, in: Intelligent Computing and Information and Communication, Springer, 2018, pp. 623–632.

[18] U. K. Sah, N. Parmar, An approach for malicious spam detection in email with comparison of different classifiers, International Research Journal of Engineering and Technology (IRJET) 4 (2017) 2238–2242.

[19] N. Jalal, A. Mehmood, G. S. Choi, I. Ashraf, A novel improved random forest for text classification using feature ranking and optimal number of trees,

Journal of King Saud University-Computer and Information Sciences (2022).
[20] C. Mi, L. Xie, Y. Zhang, Improving data augmentation for low resource speech-to-text translation with diverse paraphrasing, Neural Networks 148 (2022) 194–205.
[21] M. Kim, P. Kang, Text embedding augmentation based on retraining with pseudo-labeled adversarial embedding, IEEE Access 10 (2022) 8363–8376.
[22] S. Bonthu, A. Dayal, M. Lakshmi, S. Rama Sree, Effective text augmentation strategy for nlp models, in: Proceedings of Third International Conference on Sustainable Computing, Springer, 2022, pp. 521–531.
[23] C. Coulombe, Text data augmentation made simple by leveraging NLP cloud apis, CoRR abs/1812.04718 (2018). URL: http://arxiv.org/abs/1812.04718. arXiv:1812.04718.
[24] S. Qiu, B. Xu, J. Zhang, Y. Wang, X. Shen, G. de Melo, C. Long, X. Li, EasyAug: An Automatic Textual Data Augmentation Platform for Classification Tasks, Association for Computing Machinery, New York, NY, USA, 2020, p. 249–252. URL: https://doi.org/10.1145/3366424.3383552.
[25] Z. Feng, H. Zhou, Z. Zhu, K. Mao, Tailored text augmentation for sentiment analysis, Expert Systems with Applications (2022) 117605.
[26] R. Xiang, E. Chersoni, Q. Lu, C.-R. Huang, W. Li, Y. Long, Lexical data augmentation for sentiment analysis, Journal of the Association for Information Science and Technology 72 (2021) 1432–1447.
[27] D. T. Vu, G. Yu, C. Lee, J. Kim, Text data augmentation for the korean language, Applied Sciences 12 (2022) 3425.
[28] A. Taan, Z. Farou, Supervised learning methods for skin segmentation based on pixel color classification, Central-European Journal of New Technologies in Research, Education and Practice (2021).
[29] L. Breiman, Random forests, Machine learning 45 (2001) 5–32.
[30] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, arXiv preprint arXiv:1508.01991 (2015).
[31] K. W. Church, Word2vec, Natural Language Engineering 23 (2017) 155–162.
[32] C. Fellbaum, Wordnet, in: Theory and applications of ontology: computer applications, Springer, 2010, pp. 231–243.
[33] J. M. G. Hidalgo, T. A. Almeida, A. Yamakami, On the validity of a new sms spam collection, in: 2012 11th International Conference on Machine Learning and Applications, volume 2, IEEE, 2012, pp. 240–245.
[34] I. Cordón, S. García, A. Fernández, F. Herrera, Imbalance: oversampling algorithms for imbalanced classification in r, Knowledge-Based Systems 161 (2018) 329–341.