Text augmentation based on operation weighting using genetic algorithm

Hashem Youneszadeh Haghighi [1], Samira Noferesti *[2]

[1] Department of Information Technology Engineering, Faculty of Electrical and Computer Engineering, University of Sistan and Baluchestan, Zahedan, Iran, e-mail: h.yooneszadeh@gmail.com, +989373562524
[2] Corresponding author: Department of Information Technology Engineering, Faculty of Electrical and Computer Engineering, University of Sistan and Baluchestan, Zahedan, Iran, e-mail: snoferesti@ece.usb.ac.ir, +989151630480

Abstract

Insufficient training samples is one of the major challenges in deep learning, and one promising solution is data augmentation. Most existing methods for text data augmentation use a fixed strategy, in which some simple operations such as word replacement, insertion, deletion, and shuffling are selected randomly and applied to the text words that are also randomly sampled with equal probability. In this paper, a task-independent text augmentation approach is proposed, which, by weighting data augmentation operations using genetic algorithm, intelligently chooses the appropriate type and position of these operations for each sentences in the dataset. To evaluate the effectiveness of the proposed method, extensive experiments were conducted on several sentiment analysis datasets. In comparison with the baseline method (without data augmentation), EDA (a well-known task-independent method for text augmentation) and TTA (a state-of-the-art text augmentation method for sentiment analysis), the proposed method improves the average accuracy by 9.19%, 3.63%, and 1.04% on datasets of size 100, and by 5.27%, 3.18%, and 1.18% on datasets of size 500, respectively.

Keywords: Text augmentation, Genetic algorithm, Operation weighting, Deep learning, Sentiment analysis

Conflict of interest statement: The authors declare that they have no competing interests.

Funding sources: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

1. Introduction

In the last decade, deep learning has achieved excellent success across a wide range of applications, including natural language processing (NLP) [1-2]. One of the subfields of NLP that extensively leverages deep learning is sentiment analysis. Sentiment analysis aims to automatically classify texts into predefined categories (e.g., positive, negative, or neutral) based on emotions conveyed in the text [3].

Deep learning algorithms require massive amounts of labeled data, and insufficient training samples can result in poor performance and overfitting. Collecting and labeling training data is challenging and time-consuming, resulting in a lack of labeled training data for many applications. To tackle this

problem, a new research area called data augmentation has emerged, aiming to create synthetic training samples based on existing data.

This paper specifically addresses text augmentation that has been widely used in numerous NLP applications, including sentiment analysis, and have shown promising results [4-6]. Existing methods for text augmentation often rely on random operations or back-translation to generate new training samples. These methods have a fixed strategy for all textual inputs, while different texts may require a different set of augmentation operations. Thus, forming a data augmentation policy by determining the appropriate set of augmentation operations can be effective in enhancing the efficacy of machine learning algorithms.

In this paper, a new task-independent method for text augmentation is proposed that, rather than randomly selecting augmentation operations, intelligently determines the type and location of these operations for each input sentence utilizing a genetic algorithm. The reasons for choosing the genetic algorithm are its flexibility, adaptability, and successful results in solving various problems [7-9]. Unlike most existing approaches, the proposed method does not rely on a constant augmentation coefficient. Therefore, for each input sentence, a varying number of synthetic sentences is generated based on its characteristics, such as length and word types, which can be used to address the issue of imbalanced datasets. In addition, in the proposed method, it is possible to define a set of different augmentation operations depending on the application. This capability enables the proposed method to compete effectively with task-specific approaches.

The results of experiments conducted with a CNN classifier and on four common datasets in the field of sentiment analysis indicate that our proposed method outperforms the state-of-the-art existing methods. For instance, in comparison with Easy Data Augmentation (EDA) [10], a well-known text augmentation method, the proposed method achieves an average improvement of 3.63% and 3.18% in accuracy on datasets of size 100 and 500, respectively. Also, the average accuracy of the proposed method is about 1% higher than that of the Tailored Text Argumentation (TTA) method [4], which is a task-specific method for sentiment analysis and has achieved the best results among other compared methods.

In summary, the main contributions of this paper are as follows:

- Proposing a task-independent method for text augmentation which according to our studies, is the first attempt to intelligently choose augmentation operations
- The possibility of considering different augmentation operations depending on the application
- The possibility of considering a different augmentation coefficient for different data classes, addressing the issue of imbalanced datasets

This paper is structured as follows: Section 2 presents the literature reviews in the field of text augmentation. Section 3 describes the proposed text augmentation method. Section 4 provides the results of experiments performed to evaluate the performance of the proposed method and compares its performance with other methods. Finally, the conclusion is presented in Section 5.

2. Related Works

Current methods of text augmentation can be divided into three main categories: operation-based methods, back-translation-based methods, and combined methods. In the following, we will introduce related works in each category.

Feng et al. [11] proposed three operations for generating synthetic data: randomly swapping two adjacent characters, deleting characters, and inserting characters within words in the text. Additionally, they excluded the first and last characters of each word to more closely imitate natural noise and typos.

In [12], an approach called SSMBA is provided, which first uses a corruption function to introduce stochastic variations to data samples, then a reconstruction function backs these samples into the data domain using a pre-trained BERT model. The advantages of SSMBA are its simplicity and the lack of need for task-specific knowledge or dataset-specific fine-tuning.

In [13], a replacement-based method called CBERT is introduced. This method randomly masks some tokens in a sentence, and its objective is the prediction of a word that is compatible with the label by considering both its context and the label of the sentence.

One of the well-known methods of text augmentation is EDA [10], which augments the samples of a dataset by using four operations, including synonym replacement, random word insertion, random swap, and random word deletion. In EDA, the four mentioned operations are randomly selected and applied to each sentence in the dataset. The number of words that undergo changes is chosen proportional to the length of the sentence.

In [14], the AEDA method is introduced, which generates a random number n proportionate to the sentence length and selects n punctuation marks from {'.', ',', '!', '?', ';', ':'} to be inserted at random positions in the input sentence. In this method, the words and their order are preserved, and only their positions change.

In [4], an augmentation method for sentiment analysis called TTA is proposed. Unlike previous synonym replacement methods that consider all words in a sentence equally, TTA emphasizes the selection of words that are both discriminative and pertinent to the definition of a class. This approach consists of two primary operations: the first operation entails the probabilistic selection of a word for synonym substitution, guided by its discriminative strength and its relevance to the text's sentiment. The second operation focuses on identifying words that, while not related to the sentiment, are distinctive within the training data, and applies zero masking or contextual replacement to these words.

In [15], instead of the commonly used random deletion and synonym replacement operations for data augmentation for sentiment classification, two new operations are defined: 1) TF-IDF word dropout, and 2) adaptive synonym replacement, which help preserve the semantic and diversity of the added data.

In [16], a new text augmentation method for emotion recognition using transformers is introduced. This method replaces a word with its synonym through word embedding to achieve the same meaning but with distinct words. The technique of Bert-base-uncased pre-trained contextual word embedding is used to find word synonyms.

In back-translation, each sentence in the initial training set is first translated into a destination language and then translated back into the source language, resulting in a usually different sentence [17]. In the proposed method of [18], three types of noise are injected into the back-translated text: random word deletion, random word replacement by a filler token, and random word swapping. Another method called Tagged BT takes a different approach by considering an additional token indicating which data is original and which is synthetic [19]. Evaluation results of this method indicate its superiority compared to noise injection-based methods.

In [20], a method called back-and-forth translation is introduced. In this method, although the syntax of the translated text may vary, the underlying semantic context remains unchanged. This method has three stages: in the first stage, several sentiment classification models are created, differing solely in the size of the training and validation datasets employed to develop these models. In the second stage, sentences are translated from English to German and then reverted to English, and in the third stage, the newly generated sentences are added to the initial training set and used to train the sentiment classification models.

The main problem with operation-based methods is that the type and position of the operations for an input sentence are randomly selected. On the other hand, the problem with back-translation methods, which try to generate new sentences by translating a sentence to a target language and subsequently translating it to the source language, is that the synthetic sentences generated are repetitive or wrong in some cases. For this reason, some existing methods have utilized both techniques of operation-based and back-translation. In [21], five operations for text augmentation are applied: textual noise injection (changing, adding, deleting letters in words, changing letter case, and changing punctuation marks), spelling errors injection (generating texts with common misspellings), synonym replacement (replacing a word with one or multiple synonyms using a dictionary), paraphrase creation (through the application of regular expressions or syntactic tree structures), and back-translation.

Xie et al. [22] presented a method for text augmentation in semi-supervised learning that utilizes three techniques: RandAugment, back-translation, and substitution with TF-IDF. The third technique aims to produce a variety of valid samples by retaining keywords, which are defined as words with high TF-IDF scores, while substituting less informative words, characterized by low TF-IDF scores, with one another.

Most existing methods for text augmentation have a fixed strategy for all learning phases, and the data augmentation policy does not change during model training. In [23], dynamic policy scheduling is considered, where a search space is established for data augmentation policies, and a population-based technique is used to identify the optimal policy for each training epoch.

This paper introduces a new method for text augmentation as a pre-processing phase, which, unlike previous research that has a fixed strategy (set of operations) for all training samples, aims to intelligently select an appropriate set of operations and their positions for augmenting each sample of the training set using the genetic algorithm. Table 1 presents a summary of the related works discussed in this section, including the proposed work.

3. Material and methods

Our proposed method utilizes six basic augmentation operations, including synonym replacement, word deletion, swap, punctuation insertion, word insertion, and back-translation, with the possibility of adding more operations. The reasons for choosing these operations are their simplicity, not depending on the task and domain, and successful results in previous works. We utilize a genetic algorithm to select the appropriate operations and their suitable positions for each sentence.

Figure 1 illustrates the flowchart of the proposed method. As can be seen, the input of the genetic algorithm is a dataset of labeled sentences called *D*. Except for back translation, augmentation operations are performed only on the effective words of the text. Thus, initially, the dataset *D* is preprocessed by tokenizing and removing stop-words from its sentences. The resulted dataset is called

"Pre-processed dataset". Also, each sentence in *D* is translated into three languages, German, French, and Japanese, and a dataset named "Translated dataset" is created, which is used in back-translation.

To manage the size of the augmented dataset, the augmentation coefficient parameter (α) is defined, and the maximum size of the augmented dataset is considered to be $|D|*(\alpha+1)$. To generate the first population (generation) of the genetic algorithm, we need a dataset named D_α, in which each sentence of *D* is repeated α times. After creating D_α, the initial population is created randomly.

Each population consists of a number of individuals; with each individual representing an augmented dataset. In each iteration *t* of the genetic algorithm, for each individual (chromosome) $C_{i,t}$, the appropriate operations and their positions to augment each sentence of the dataset *D* are determined using genetic operations, including selection, crossover, and mutation. The class label of each synthetic sentence is the same as the original sentence. The augmented dataset $D_{i,t}$ created based on the individual $C_{i,t}$ is used as the training set for a CNN classifier. The CNN is trained on $D_{i,t}$ to predict the class labels for the validation samples (*V*). If the accuracy of the CNN classifier for dataset $D_{i,t}(Acc_{i,t})$ exceeds the best accuracy obtained so far $(Best_{Acc})$, it indicates that this augmented dataset $(D_{i,t})$ is better than those produced in previous iterations. This process continues until the termination condition of the genetic algorithm is satisfied. Finally, the dataset that leads to the highest accuracy of the CNN classifier is chosen as the final solution.

3.1 Problem Encoding

In the proposed method for text augmentation, binary encoding is used. As shown in Figure 2, each individual is a three-dimensional array (Ar), where the first dimension of the array represents the sentences in the dataset, the second dimension represents the augmentation operations for each sentence, and the third dimension represents the words in that sentence. Each element of the array can take a value of one or zero, indicating whether the corresponding operation should be applied to the corresponding word in the target sentence. With the above description, individual *i* in generation *t* (denoted as $C_{i,t}$) is defined as follows:

$$C_{i,t} = Ar[S,O,W] \tag{1}$$

where *S* is the maximum number of synthetic sentences of the augmented dataset, *O* is the number of augmentation operations, and *W* is the number of words in a sentence (after removing stop words). The value of *S* in Equation (1) is considered as $S = |D|*\alpha$. However, some of the generated synthetic sentences are repetitive and also in some cases, all the genes of an individual become zero, which means no new sentence is produced. For this reason, the number of synthetic sentences will not be exactly *S*. In this paper, the value of α is considered equal to 4, but it is possible to adjust the value of α according to the size of the initial dataset.

In Equation (1), $C_{i,t}[j][k][z]=1$ indicates that the *k*th operation should be applied to the *z*th word in the *j*th sentence of dataset D_α to create a new sentence. As mentioned before, D_α contains the sentences of dataset *D*, each sentence is repeated α times. In other words, the first α sentences of D_α

are equal to the first sentence of *D*, the second α sentences of D_α are equal to the second sentence of *D*, and this process is repeated for other sentences.

Figure 2 illustrates an example of individual encoding in the proposed approach. In this example, for the first sentence in the dataset D_α, the "random word deletion" operation is applied to the first word, and the "random swap" operation is applied to the third word.

3.2 The initial population

The initial population (first generation), denoted as P_1, is created randomly. More precisely, the first population consists of *N* individuals, and each individual is represented by a three-dimensional binary array, where the elements of the array are randomly assigned values of one with a probability of *p* percent of the length of the corresponding sentence.

3.3 Generating the augmented dataset and evaluating the fitness of individuals

At the end of each iteration *t* of the genetic algorithm, the array of individual $C_{i,t}$ is passed to a function. In this function, a new sentence is generated based on the selected operations (the ones in the array) for each sentence in D_α. For this purpose, six augmentation operations including random synonym replacement, random word deletion, random swap, random punctuation insertion, random word insertion, and back-translation, are encoded as numbers 0 to 5, respectively. Then, to convert the individual $C_{i,t}$ into the augmented dataset $D_{i,t}$, the following steps are performed:

- Random synonym replacement: If $C_{i,t}[j][0][z]=1$, a random number from the set {1,2} is chosen. If 1, a synonym for the *z*th word in sentence S_j is found using WordNet [24], and replaces the original word. If 2, two synonyms are found and used to replace the original word.
- Random word deletion: If $C_{i,t}[j][1][z]=1$, the *z*th word will be deleted from sentence S_j.
- Swapping: If $C_{i,t}[j][2][z]=1$, the *z*th word in sentence S_j is swapped with another randomly selected word from the same sentence.
- Punctuation insertion: If $C_{i,t}[j][3][z]=1$, a punctuation mark is randomly selected from the set {'.', ',', '!', '?', ';', ':'}, and it is inserted instead of the *z*th word in sentence S_j.
- Random insertion: If $C_{i,t}[j][4][z]=1$, using the WordNet, one of the synonyms for the *z*th word is selected and inserted after the *z*th word in sentence S_j.
- Back translation: If $C_{i,t}[j][5][z]=1$, the target language for translation is chosen based on the position *z* within the sentence S_j: German for the first third, French for the middle third, and Japanese for the final third. The sentence S_j is then translated into the target language and re-translated back into English.

At the end of iteration *t* of the genetic algorithm, the dataset *D*, and newly generated sentences are combined into $D_{i,t}$, with duplicates removed. The dataset $D_{i,t}$, along with the validation set, is provided

to the CNN classifier. The CNN's accuracy for the augmented dataset $D_{i,t}(Acc_{i,t})$ indicates the fitness of individual $C_{i,t}$.

3.4 Generating the next generations

To create the next generation, genetic operators such as selection, crossover, and mutation are applied. After applying the genetic operators, four offspring with the highest fitness are guaranteed to be included, while the remaining individuals are selected from the best individuals in list *L*, which includes both offspring and parents.

3.4.1 Selection

First, the individuals of the current population (P_t) are sorted in descending order based on their fitness. Then, half of the individuals are removed from the end of the list, and parents are randomly selected from the remaining individuals.

3.4.2 Crossover

The two selected parents exchange information using the two-point crossover method, where two randomly generated points in the sentence are used to swap segments and create new offspring. This process, carried out with a probability *pc*, results in two new individuals.

3.4.3 Mutation

The mutation operator reverses the values of some genes in individuals with a small probability *pm*.

3.5 Termination condition

The genetic algorithm terminates when it reaches a specified number of generations (Max_t) or when the best accuracy of the CNN does not change for 35 consecutive iterations of the genetic algorithm.

4. Results

In this section, first, the simulation settings for the proposed method are presented. Then, the datasets used are described. Finally, the effectiveness of our proposed method is assessed and compared with the current methods.

4.1 Simulation settings

The experimental environment used in this study includes Intel(R) Corei7, 7th Generation with 16 GB RAM, Windows 10 OS. The code was implemented using Python version 3.9.13 and the deep learning framework PyTorch 3.19.13. The parameters of the genetic algorithm and the variables defined in our method were set according to Table 2. The characteristics of the CNN classifier are also provided in Table 3.

4.2 Datasets

To evaluate the performance of the proposed method, we used four commonly used datasets in sentiment analysis: SST-2, Sentiment 140, Yelp, and US Airline. Table 4 presents the details of each dataset.

Each dataset is divided into three parts: train, validation, and test. The training set is the initial dataset *D*, which is used as input to the proposed method. To create the test and validation sets, random sampling was used [4]. The validation set was determined to be half the size of the training set, while the test set was selected to be as large as possible. The sizes of these datasets are presented in Table 5.

4.3 Evaluating the proposed method

Table 6 presents the number of sentences generated by our method for each dataset described in Table 4. For example, the 100-sentence SST-2 dataset was augmented to consist of 445 sentences, with 226 positive labels and 219 negative labels. Similarly, the 500-sentence SST-2 dataset was increased to 2461 sentences, with 1209 positive labels and 1252 negative labels. On average, the examined datasets of 100 and 500 sentences increased in size by 4.42 and 4.64 times, respectively.

Table 7 shows some synthetic sentences produced by the proposed method.

Table 8 presents the generated sentences related to one sentence. As can be seen, four synthetic sentences have been produced for the original sentence.

To evaluate the performance of the proposed method, the augmented dataset resulting from applying the proposed method to the initial dataset (each of the training sets in Table 5) was used as the training set for a CNN classifier. The classifier's accuracy was then calculated on the test set. Subsequently, the accuracy of the CNN classifier was compared with its accuracy when trained on the initial dataset without data augmentation (No-DA), as well as on datasets augmented by several existing methods, including Back-translation [18], EDA [10], CBERT [13], TF-IDF replacing [22], SSMBA [12], BF-Translation [20], and TTA [4], as introduced in Section 2. Accuracy represents the proportion of correctly predicted model outcomes to the total number of model predictions. We calculated the accuracy percentage using the following formula:

$$Accuracy = (Number\ of\ correct\ predictions) / (Total\ number\ of\ predictions) * 100 \quad (2)$$

Experimental results in table 9 show that our approach has significantly improved accuracy compared to the baseline method without augmentation. On average, for the 100- and 500- sentence datasets, the proposed method has improved the accuracy of the baseline by 9.19% and 5.26%, respectively.

As observed in Table 9, the proposed method outperformed the compared methods in 5 out of 6 evaluations (except for the 500-sentence US Airline dataset). The last row of table 9 indicates the improvement in the performance of the CNN classifier when using the proposed method for text augmentation compared to the TTA method, which performed better than the other methods. For instance, in the 500-sentence Senti140 dataset, the proposed method achieved a 3.29% improvement in accuracy compared to TTA.

Figures 3 and 4 compare the average accuracy of the proposed method on the 100- and 500-sentence datasets, respectively, with that of the other methods. As observed, the proposed method outperforms all other methods in terms of average accuracy. In comparison with EDA and Back-translation, two well-known task-independent methods, our proposed method improves average accuracy by 3.63% and 5.36% on the 100-sentence dataset, and by 3.18% and 3.4% on the 500-sentence dataset, respectively.

The proposed method exhibited an approximately 1% increase in average accuracy compared to TTA for the 100-sentence dataset and around a 1.18% increase for the 500-sentence dataset. It should be noted that the proposed method uses general augmentation operations and thus can be easily applied to different kinds of text classification, while TTA is task-specific and specifically introduced to improve the performance of sentiment classification. Therefore, it is expected that the performance of sentiment classification will improve by employing task-specific augmentation operations, such as those introduced in [4], in the proposed method, which we will address as future work.

5. Conclusions

In this paper, we proposed an intelligent text augmentation method using a genetic algorithm that generates synthetic sentences based on the characteristics of each sentence. The main advantages of the proposed method compared to existing methods are: 1) intelligently selecting the type and position of augmentation operations based on the characteristics of the text words; 2) using a non-constant augmentation coefficient that results in generating a varying number of synthetic sentences for each input sentence based on its characteristics, such as length and word types; and 3) the possibility of using general-purpose augmentation operations or a different set of operations depending on the application. The experimental results indicate that our proposed method achieves greater accuracy than the baseline method (without data augmentation), task-independent existing methods, and state-of-the-art text augmentation methods for sentiment analysis.

Future research endeavors will focus on applying the proposed method to various NLP tasks and domains, optimizing augmentation coefficients specifically tailored for imbalanced datasets, integrating the proposed method with other machine learning techniques, developing better evaluation metrics and benchmarking, as well as conducting parameter tuning and sensitivity analysis.

References

[1] Alkishri, W., Widyarto, S., Yousif, J. H., et al. “Fake Face Detection Based on Colour Textual Analysis Using Deep Convolutional Neural Network”, *J Internet ServvnInf Secur,* **13**(3), pp. 143-155 (2023). https://doi.org/10.58346/jisis.2023.i3.009.

[2] Goyal, P., Pandey, S. and Jain, K. “Deep learning for natural language processing”, *New York: Apress*, 2018. https://doi.org/10.1007/978-1-4842-3685-7.

[3] Liu, B. “Sentiment analysis and opinion mining”, *Synthesis Lectures on Human Language Technologies*, **5**(1), pp. 1-167 (2012). https://doi.org/10.1007/978-3-031-02145-9.

[4] Feng, Z., Zhou, H., Zhu, Z., et al. “Tailored text augmentation for sentiment analysis”, *Expert Systems with Applications,* **205**, pp. 117605 (2022). https://doi.org/10.1016/j.eswa.2022.117605.

[5] Pellicer, L. F., Ferreira, M. T. and Costa, A. H. “Data augmentation techniques in natural language processing”, *Applied Soft Computing,* **132**, pp. 109803 (2023). https://doi.org/10.1016/j.asoc.2022.109803.

[6] Li, B., Hou, Y. and Che, W. “Data augmentation approaches in natural language processing”, *AI Open,* **3**, pp. 71-90 (2022). https://doi.org/10.1016/j.aiopen.2022.03.001.

[7] Fu, Q. W., Liu, Q. H. and Hu, T. “Multi-objective optimization research on VR task scenario design based on cognitive load”, *Facta Universitatis, Series: Mechanical Engineering*, **22**, pp. 293-313 (2024). https://doi.org/10.22190/FUME240122029F.

[8] Mzili, T., Mzili, I., Riffi, M. E., et al. “Hybrid genetic and penguin search optimization algorithm (GA-PSEOA) for efficient flow shop scheduling solutions”, *Facta Universitatis, Series: Mechanical Engineering*, **22**, pp. 077-100 (2024). https://doi.org/10.22190/FUME230615028M.

[9] Soori, M. and Asmael, M. “Minimization of deflection error in five axis milling of impeller blades”, *Facta Universitatis, series: Mechanical Engineering*, **21**, pp. 175-190 (2023). https://doi.org/10.22190/FUME210822069S.

[10] Wei, J. and 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)*, Association for Computational Linguistics, China, pp. 6382–6388 (2019). https://doi.org/10.48550/arXiv.1901.11196.

[11] Feng, S. Y., Gangal, V., Kang, D. et al. “Genaug: Data augmentation for finetuning text generators”, *In Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 29–42 (2020). https://doi.org/10.18653/v1/2020.deelio-1.4. (2020). https://doi.org/10.48550/arXiv.2010.01794.

[12] Ng, N., Cho, K. and Ghassemi, M. “SSMBA: Self-supervised manifold based data augmentation for improving out-of-domain robustness”, *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, pp. 1268–1283 (2020). https://doi.org/10.48550/arXiv.2009.10195.

[13] Wu, X., Lv, S., Zang, L., et al. “Conditional BERT Contextual Augmentation”, *Computational Science–ICCS 2019: 19th International Conference*, Portugal, pp. 84-95 (2019). https://doi.org/10.48550/arXiv.1812.06705.

[14] Karimi, A., Rossi, L. and Prati, A. “AEDA: An Easier Data Augmentation Technique for Text Classification”, *Findings of the Association for Computational Linguistics: EMNLP 2021*, Association for Computational Linguistics, pp. 2748–2754 (2021). https://doi.org/10.18653/v1/2021.findings-emnlp.234.

[15] Chao, G., Liu, J., Wang, M., et al. “Data augmentation for sentiment classification with semantic preservation and diversity”, *Knowledge-Based Systems*, **280**, pp. 111038 (2023). https://doi.org/10.1016/j.knosys.2023.111038.

[16] Mohammad, F., Khan, M., Marwat, S. N., et al. “Text Augmentation-Based Model for Emotion Recognition Using Transformers”, *Computers, Materials & Continua*, **76**, pp. 3523-3547 (2023). https://doi.org/10.32604/cmc.2023.040202.

[17] Hoang, C., Koehn, P., Haffari, G., et al. (2018). Iterative back-translation for neural machine translation. *In 2nd Workshop on Neural Machine Translation and Generation.* Association for Computational Linguistics, Australia, pp. 18-24, https://doi.org/10.18653/v1/W18-2703.

[18] Edunov, S., Ot, M., Auli, M., et al. “Understanding back-translation at scale”, *In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,* Association for Computational Linguistics, Belgium, pp. 489-500 (2018). https://doi.org/10.18653/v1/D18-1045.

[19] Caswell, I., Chelba, C. and Grangier, D. “Tagged back-translation”, *In Proceedings of the Fourth Conference on Machine Translation,* Association for Computational Linguistics, Italy, pp. 53-63 (2019). https://doi.org/10.18653/v1/W19-5206.

[20] Body, T., Tao, X., Li, Y., et al. “Using back-and-forth translation to create artificial augmented textual data for sentiment analysis models”, *Expert Systems with Applications,* **178**, pp. 115033 (2021). https://doi.org/10.1016/j.eswa.2021.115033.

[21] Coulombe, C. “Text data augmentation made simple by leveraging nlp cloud apis”, *ArXiv, abs/1812*, 04718 (2018). https://doi.org/10.48550/arXiv.1812.04718.

[22] Xie, Q., Dai, Z., Hovy, E., et al. “Unsupervised Data Augmentation for Consistency Training”, *Advances in neural information processing systems*, **33**, pp. 6256-6268 (2020). https://doi.org/10.48550/arXiv.1904.12848.

[23] Li, S., Ao, X., Pan, F., et al. “Learning policy scheduling for text augmentation”, *Neural Networks,* **145**, pp. 121-127 (2022). https://doi.org/10.1016/j.neunet.2021.09.028.

[24] Miller, G. A. “WordNet: a lexical database for English”, *Communications of the ACM,* **38**(11), pp. 39–41 (1995). https://doi.org/10.1145/219717.219748.

Figure and table captions

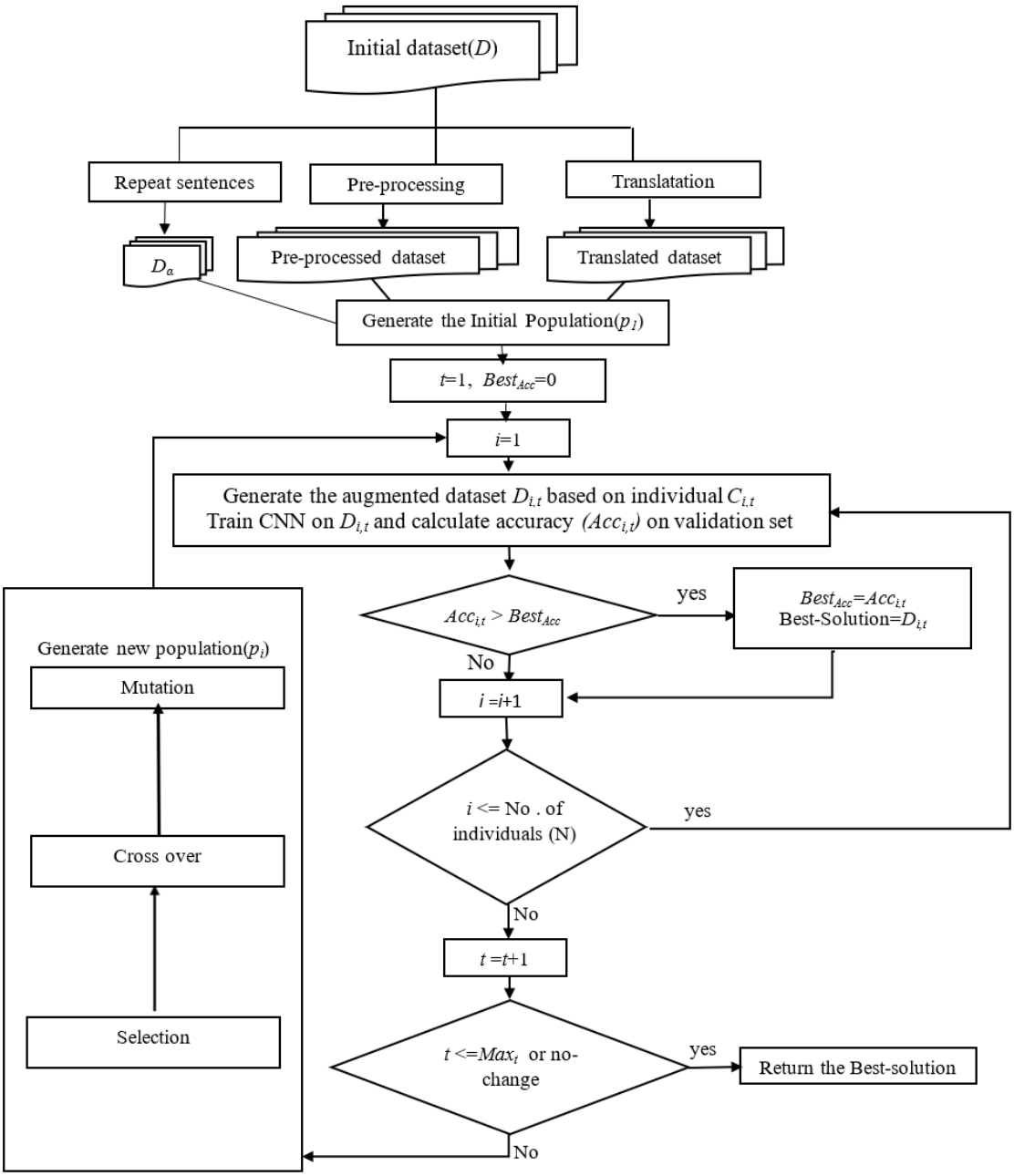
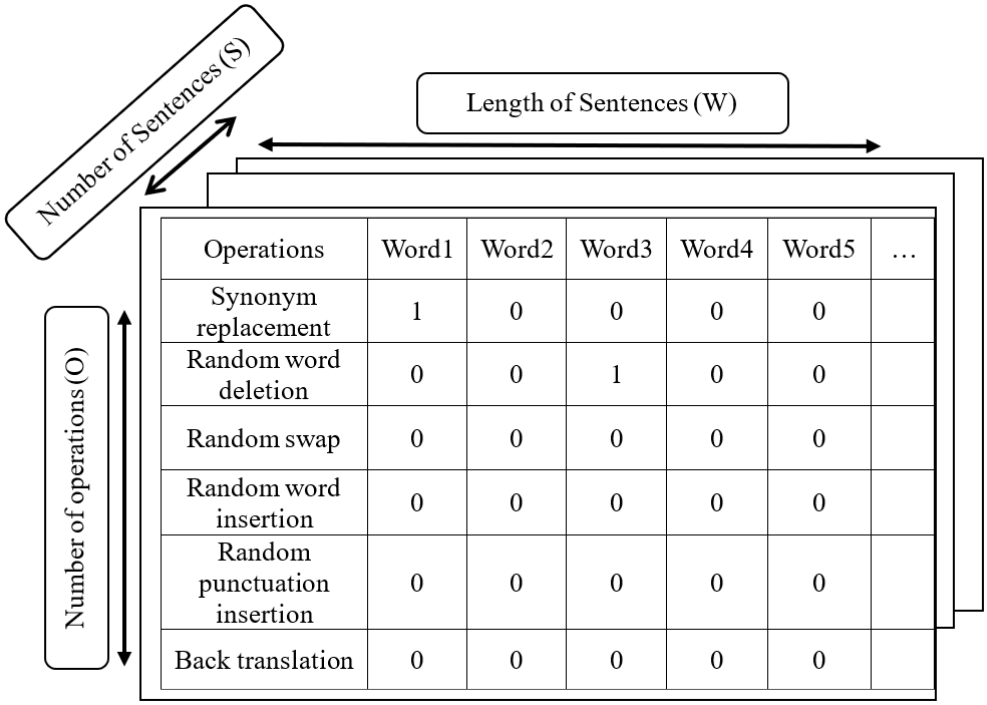


Figure 1. Flowchart of the proposed method for text augmentation



Operations	Word1	Word2	Word3	Word4	Word5	…
Synonym replacement	1	0	0	0	0	
Random word deletion	0	0	1	0	0	
Random swap	0	0	0	0	0	
Random word insertion	0	0	0	0	0	
Random punctuation insertion	0	0	0	0	0	
Back translation	0	0	0	0	0	

Figure 2. Example of individual encoding in the genetic algorithm

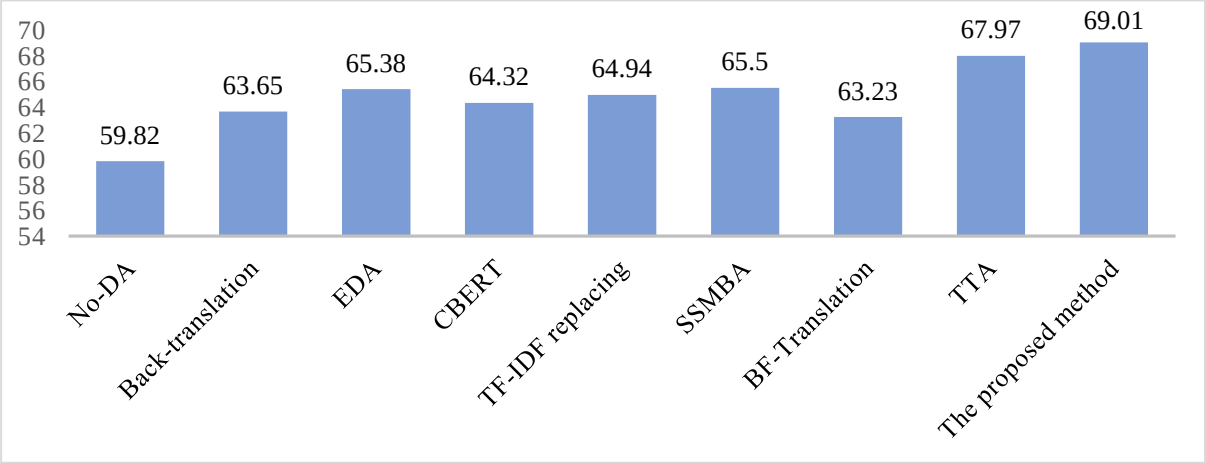


Figure 3. Average accuracy of the proposed method and other methods on the 100-sentence datasets

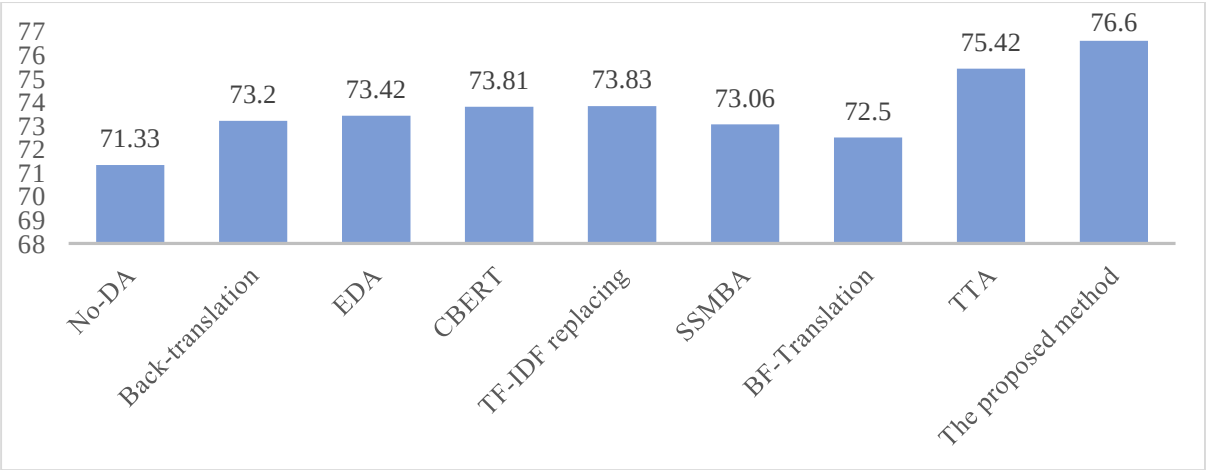


Figure 4. Average accuracy of the proposed method and other methods on the 500-sentence datasets

Table 1. Summary of the existing works

Reference	Category	Features	Task-dependent	Fixed-strategy
[10-14]	Operation-based	Randomly selecting augmentation operations and their positions	×	√
[4, 15, 16]	Operation-based	Selecting only discriminative and pertinent words	√	√
[19]	Back-translation-based	Translating text to another language and re-translating to the initial language	×	√
[20]	Back-translation-based	Translating text to another language and re-translating to the initial language	√	√
[18, 21, 22]	Combined	Using both back-translation and augmentation operations	×	√
[23]	Population-based	Dynamic policy scheduling	×	×
The proposed method	Combined	Using a genetic algorithm to select the appropriate operations and their suitable positions	×	×

Table 2. Parameters used for simulation

Parameter	Symbol	Value
No. of generations	Max_t	100-200
Population size	N	50-100
No. of operations	O	6
Augmentation coefficient	α	4
Crossover probability	pc	0.45
Mutation probability	pm	0.01
The probability of a gene becoming 1 in generation P_1	p	0.05

Table 3. CNN characteristics

The number of layers	4
First layer	Embedding Layer with 300-dimensional vectors using GloVe 42B.300d
Second layer	Convolutional layer including 100 filters 2×3×4
Third layer	Max-pooling layer with size 2×2
Fourth layer	Fully Connected Layer
Optimizer	Adam
Cost function	Cross-Entropy Loss
Learning rate	0.0001
Dropout rate	0.5
Batch size	64
No. of epochs	20

Table 4. Specifications of the datasets

Dataset	Labels	#Sentences	Year	Description
SST-2	Positive/Negative	11855	2013	It is a collection of movie reviews.
Sentiment 140	Positive/Negative	1600000	2009	It allows you to discover the sentiments of a brand, product, or topic on Twitter.
Yelp	Positive/Negative	5600	2015	This dataset is a subset of Yelp's businesses, reviews, and user data.
US Airline	Positive/Negative/Neutral	14640	2015	This dataset contains tweets about major US airlines.

Table 5. Statistics of the training, validation and test sets [4]

	Us Airline		Yelp		Senti140		SST2	
Training set	500	100	500	100	500	100	500	100
Validation set	250	50	250	50	250	50	250	50
Test set	6778	6778	10000	10000	10000	10000	2000	2000

Table 6. Size of the original and augmented datasets

Dataset	Original	Augmented	Positive labels	Negative labels	Neutral labels
SST2	100	445	226	219	--
	500	2461	1209	1252	--
Senti140	100	439	214	225	--
	500	2301	1162	1139	--
Yelp	100	438	220	218	--
	500	2211	1069	1142	--
US Airline	100	446	153	153	140
	500	2311	781	762	768
Average	100	442	Augmentation ratio is 4.42		
	500	2321	Augmentation ratio is 4.64		

Table 7. Sample sentences generated by the proposed method

	Original sentence	Synthetic sentence	Operations
1	though lan yu lacks a sense of dramatic urgency , the film makes up for it with a pleasing verisimilitude .	though lan yu lacks a sense of dramatic **striking** urgency , the film makes up for it with a pleasing verisimilitude .	Synonym replacement
2	For those with the stomach and stamina for its heartbeat-quickening intensity and body-slamming action , rollerball delivers exactly what it promises : a people 's hero you can really get behind .	For those with stomach and perseverance for his heartbeat intensity and the action of bodily harm, rollerball provides exactly what it promises: a hero of the people that you really get behind.	Back translation
3	his good looks , charm and overwhelming confidence captured the eye of screen legend norma shearer , who offered him a film role .	his good looks , charm and overwhelming confidence captured the eye of ~~**screen**~~ legend norma shearer , who offered him a film role .	Random deletion
4	after repeated trips to cuba , the schendel brothers succeed in taking a close look into the underground world of cuban cars , finding along the way a gallery of eccentric characters - the curators of the largest , living , automobile museum in the world .	after repeated trips to cuba , the schendel brothers succeed in taking a close look into the underground **way** of cuban cars , finding along the **world** a gallery of eccentric characters - the curators of the largest , living , automobile museum in the way .	Random swap
5	now , in present day , washed-up child actor julian (luke eberl) , the free-spirited hannah (colombe jacobsen) , and former teacher grace (jenny mollen) , along with a documentary film crew , go on a search for this man , who may be the embodiment of all evil .	now , in present day , washed-up child actor julian (luke eberl), the free-spirited hannah (colombe jacobsen), and former teacher grace (~~**jenny**~~ mollen) , along with a documentary film crew , go on a search for this man , who may be the embodiment of all **malign**.	Synonym replacement, Random deletion

Table 8. Sentences generated for an example sentence

Original sentence: [villeneuve] seems to realize intuitively that even morality is reduced to an option by the ultimate mysteries of life and death	
Operation	**Synthetic sentence**
Random swap	[villeneuve] **death** to realize intuitively that even morality is reduced to an option by the ultimate mysteries of life and **seems**
Random deletion	[villeneuve] seems to realize intuitively that even morality is reduced to an option by the ~~**ultimate**~~ mysteries of life and death
Synonym replacement	[villeneuve] seems to realize intuitively that even morality is reduced to an option by the ultimate **mystery** of life and death
Random swap	[villeneuve] seems to realize intuitively that even morality is **option** to an **reduced** by the ultimate mysteries of life and death

Table 9. Comparison of the the proposed method with existing methods in terms of accuracy

Dataset	**US Airline**		**Yelp**		**Senti140**		**SST-2**	
Size	500	100	500	100	500	100	500	100
No-DA	58.62	48.08	79.45	69.31	61.21	55.87	86.05	66.00
Back-translation	59.90	53.07	80.97	69.33	64.41	57.13	87.50	75.05
EDA	61.12	58.47	80.79	69.11	64.76	57.07	87.00	76.85
CBERT	61.40	55.83	80.24	69.32	66.21	57.34	87.40	74.80
TF-IDF replacing	63.65	58.97	79.94	68.80	64.29	57.70	87.45	74.30
SSMBA	61.77	55.50	79.21	69.76	64.17	57.00	87.10	79.75
BF-Translation	61.64	53.39	78.83	69.23	62.29	56.06	87.25	74.25
TTA	65.51	61.85	81.15	71.18	66.71	58.98	88.30	79.85
The proposed method	65.33	63	82.18	71.50	70	61.08	88.87	80.45
Improvement compared to TTA (%)	-0.18	1.15	1.03	0.32	3.29	2.1	0.57	0.6

Biographies

Hashem Youneszadeh Haghighi received his bachelor's degree in Computer Engineering in 2012 from Jahrom University, Iran. He is currently a graduated Master's student in Information Technology Engineering at the University of Sistan and Baluchestan, Zahedan, Iran. His research interests include machine learning and data mining.

Samira Noferesti is currently an Associate Professor in the Department of Electrical and Computer Engineering at the University of Sistan and Baluchestan, Zahedan, Iran. She completed her bachelor's degree in Computer Engineering at Sharif University of Technology in Tehran, Iran, in 2003. She received her master's degree in Computer Software Engineering from AmirKabir University of Technology in 2005. She obtained her PhD in Computer Software Engineering from Shahid Beheshti University in 2015. Her primary fields of interest include artificial intelligence, natural language processing, and opinion mining.