IJWR

Learning an Efficient Text Augmentation Strategy: A Case Study in Sentiment Analysis

Mehdy Roayaei*
Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran, mroayaei@modares.ac.ir

ABSTRACT

Contemporary machine learning models, like deep neural networks, require substantial labeled datasets for proper training. However, in areas such as natural language processing, a shortage of labeled data can lead to overfitting. To address this challenge, data augmentation, which involves transforming data points to maintain class labels and provide additional valuable information, has become an effective strategy. In this paper, a deep reinforcement learning-based text augmentation method for sentiment analysis was introduced, combining reinforcement learning with deep learning. The technique uses Deep Q-Network (DQN) as the reinforcement learning method to search for an efficient augmentation strategy, employing four text augmentation transformations: random deletion, synonym replacement, random swapping, and random insertion. Additionally, various deep learning networks, including CNN, Bi-LSTM, Transformer, BERT, and XLNet, were evaluated for the training phase. Experimental findings show that the proposed technique can achieve an accuracy of 65.1% with only 20% of the dataset and 69.3% with 40% of the dataset. Furthermore, with just 10% of the dataset, the method yields an F1-score of 62.1%, rising to 69.1% with 40% of the dataset, outperforming previous approaches. Evaluation on the SemEval dataset demonstrates that reinforcement learning can efficiently augment text datasets for improved sentiment analysis results.

***Keywords*— Data Augmentation, Sentiment Analysis, Deep Reinforcement Learning, Neural Network, DQN Algorithm.**

1. Introduction

In the recent past, there has been a notable rise in the application of deep learning methodologies. Contemporary machine learning models, including deep neural networks, often involve numerous parameters, necessitating substantial labeled datasets for effective training. The size and quality of the training set significantly influence the outcomes of deep learning tasks. Consequently, data assumes a pivotal role in determining the performance of deep neural networks. Nonetheless, the process of collecting and labeling an extensive set of real-world samples is frequently costly, error-prone, and existing datasets commonly exhibit imbalances in data distribution.

The method of artificially enlarging labeled training sets, through the transformation of data points while maintaining class labels—referred to as *data augmentation*—has rapidly emerged as a crucial and effective solution for addressing the issue of insufficient labeled data [1]. The motivation for employing data augmentation is to address the constraints posed by limited data, inadequacies in data volume, or imbalanced data distribution. These challenges can contribute to issues like overfitting and over-parameterization, ultimately diminishing the efficacy of the learning outcomes.

Consequently, different data augmentation methods have been extensively employed to enhance the training dataset [2]. As an example, convolutional deep neural networks have demonstrated proficiency in numerous machine vision tasks.

Unfortunately, numerous application domains, including medical image analysis and sentiment analysis, lack access to large datasets [3]. Data Augmentation serves as a strong approach to address this limitation. By generating augmented data, a more comprehensive set of potential data points is represented, thereby reducing the gap between the

doi http://dx.doi.org/10.22133/ijwr.2024.441414.1202
Citation M. Roayaei, "Learning an Efficient Text Augmentation Strategy: A Case Study in Sentiment Analysis", *International Journal of Web Research*, vol.6, no.2,pp.67-75, 2023, doi: http://dx.doi.org/10.22133/ijwr.2024.441414.1202.

*Coressponding Author
Article History: Received: 3 October 2023 ; Revised: 17 November 2023; Accepted: 24 November 2023.

training and validation sets, as well as any forthcoming testing sets. This is performed on the assumption that augmentations enable the extraction of additional information from the original dataset [4].

Data augmentation has found extensive applications in enhancing the performance of deep neural networks across diverse domains, notably in computer vision. In the realm of computer vision, diverse techniques are available for artificially creating such data. For images, transformations such as rotation or adjusting the RGB channel prove beneficial, aiming to maintain model invariance to these changes. Similarly, in the domain of speech recognition, methods involving alterations to sound or speed are commonly employed. Conversely, the exploration of data augmentation in natural language processing (NLP) encounters the challenge of formulating general guidelines for transforming textual data. This challenge stems from the need to ensure that automated transformations preserve the quality of labeling [5, 6].

Recently, data augmentation has advanced significantly in the field of NLP, primarily due to increased work in domains with popular neural networks that require a huge amount of training data [7, 8]. In spite of these recent advancements, this area is still relatively new and has not been extensively studied, mainly due to the challenges arising from the complex nature of language data [9].

Reinforcement Learning (RL) is a powerful approach within the field of machine learning. At its core, RL is concerned with the development of intelligent agents capable of learning how to make sequences of decisions that optimize their performance in a given environment. In contrast to supervised learning, which utilizes labeled data for training, and unsupervised learning, where algorithms independently discover patterns, RL involves agents that learn through interaction. These agents receive feedback, manifesting as rewards or penalties based on their actions, and their primary goal is to acquire a policy—a strategy or decision-making process—that maximizes the cumulative reward over time. This approach has found applications in a wide range of fields, from robotics to games, and even in the data augmentation [1], making RL a versatile and compelling area of research and application.

RL has recently found application in augmenting text data to enhance classifier performance, particularly in situations involving limited data. An example of such a framework is Data Boost, employing RL-guided conditional generation for data augmentation [10]. Data Boost has been evaluated on three different text classification tasks involving five distinct classifier architectures. The outcomes indicate that, on average, it enhances the F1-score of classifiers by 8.7% when provided with only 10% of the entire dataset for training. Furthermore, evaluations verify that the augmentation introduced by Data Boost maintains a comparable level of quality to the original data in terms of readability and class consistency. We have chosen Data Boost as the primary baseline due to its utilization of RL for text data augmentation, its evaluation in sentiment analysis, and its use of the same dataset as employed in this paper.

The aim of this paper is to develop an effective text data augmentation strategy using a deep reinforcement learning method, called RLAug2Sent, to enhance accuracy in sentiment analysis. In NLP, sentiment analysis entails examining and identifying the sentiment or emotional tone conveyed in text. This sentiment is commonly categorized as negative, positive, or neutral, providing insights into people's opinions or attitudes expressed within the text.

We employ four different text data transformation transformations and learn how to apply these transformations to the original datasets in a manner that yields augmented datasets capable of improved sentiment classification. Our contributions can be stated as follows:

- Generating new textual data by combining multiple data augmentation transformations.
- Defining a data augmentation policy using multiple transformations.
- Finding an efficient data augmentation policy using deep reinforcement learning.
- Utilizing DQN (Deep Q-Network) network for solving the text data augmentation problem.
- Evaluating and comparing the proposed method on SemEval sentiment analysis dataset.

The remainder of this paper is organized as follows. Section 2 summarizes previous works. Section 3 introduces the proposed method. Section 4 includes results and discussion. Finally, we conclude the paper in Section 5.

2. Previous Works

Data augmentation refers to a collection of algorithms that generate artificial data from an existing dataset. These artificially created data usually contain small variations or modifications in the original data. Data augmentation stands out as a highly beneficial technique for shaping the training of deep neural networks. Its primary application lies in preventing overfitting, a situation in which a model becomes excessively tailored to the training data, resulting in suboptimal performance on novel, unseen data.

Shorten et al. [11] focused on research that explores the use of augmentations to create test sets that promote generalization. In NLP, the application of data augmentation is still in its early stages compared to computer vision. Therefore, they highlighted the substantial distinctions between the two domains and considered promising ideas that are yet to undergo comprehensive testing in the field of NLP.

Raille et al. [12] presented Edit Transformer, a neural architecture that generates new sentences by incorporating relevant edits. The key advancement compared to previous approaches is the Edit Transformer's capability to perform cross-domain editing. Furthermore, they demonstrated that the Edit Transformer offers faster training and usage compared to its predecessor.

Generally, data augmentation techniques can be categorized into four groups: *paraphrasing-based*, *noising-based*, *sampling-based*, and *reinforcement learning-based*, considering the varied nature of the introduced data [13].

Paraphrases frequently occur in natural language, offering alternative expressions to convey the same information as the original form [14, 15, 16, 17]. Generating paraphrases naturally lends itself as a suitable approach for data augmentation. It can take place at various levels, such as lexical paraphrasing, phrase paraphrasing, and sentence paraphrasing.

The objective of paraphrasing is to ensure that the semantics of the augmented data closely resemble those of the original data. On the other hand, noising-based methods introduce subtle noise that does not significantly impact the semantics, allowing the augmented data to deviate appropriately from the original data. The commonly used noises for text augmentation are swapping (random exchange of words, and even entire sentences, within a sensible range) [6, 18], deletion (randomly removing words from a sentence or deleting sentences within a document) [19, 20], insertion (randomly inserting words into a sentence or adding sentences to a document) [21, 22], and substitution (randomly substituting words or sentences with different strings) [23, 24].

Sampling-based techniques comprehend the inherent data distribution and produce new data samples originating from that distribution. Similar to models centered on paraphrasing, these techniques employ rules and trained models for creating augmented data. Nonetheless, the principal distinction lies in the task specificity of sampling-based methods, as they depend on task-related information like labels and data format. These methods not only guarantee the validity of the generated data but also elevate diversity by introducing various samples from the data distribution. Sampling methods include rule-based method [25], non-pretrained model [26], pre-trained model [27], self-training [28], and mix-up [29].

Liu et al. [10] proposed a framework for augmenting text data, known as Data Boost, which utilizes reinforcement learning-guided conditional generation. They used the GPT-2, and took advantage of policy gradient method for policy optimization in reinforcement learning.

Yu et al. [30] used inverse reinforcement learning to propose a flexible sentence-level text augmentation technique to improve model performance. They demonstrated the ability to manage the generated data by establishing evaluation metrics specific to the task. Those metrics were used to rank the self-augmented data.

Fang et al. [31] leveraged reinforcement learning to significantly expand the training dataset and proficiently identify coreference event mentions.

Kim et al. [32] explored the application of RL to enhance learning in text-based games. By leveraging methods such as synonym replacement, word permutation, and character masking, the study aims to improve the generalization and robustness of models trained on limited data. Through experiments and analysis, they demonstrated how these augmentation strategies can effectively boost performance and adaptability in text-based game playing agents, showcasing the potential of data augmentation in enhancing learning outcomes in complex environments.

3. Proposed Method

This paper presents a method for automatic search of data augmentation strategy using deep reinforcement learning, named **RLAug2Sent**. Each augmentation strategy is considered as a policy of reinforcement learning. The policy space is designed such that each policy consists of a vector of length equal to the number of data augmentation transformations (4, in this case), indicating the probability of applying each type of data augmentation transformation on the original dataset. A search algorithm is employed to find the best policy (best combination of probabilities), ensuring that the training neural network achieves the highest accuracy on the target dataset. The details of **RLAug2Sent** will be discussed further.

3.1. Structure of RLAug2Sent

Figure 1 outlines the steps of the proposed method. It constitutes two neural networks (the *policy network* and the *training network*) and four major steps. The policy network is employed to learn the augmentation policy (strategy), while the training network is used to train NLP task (sentiment analysis, in this case) on the augmented dataset. Each policy, which represents an augmentation strategy, employs a combination of text augmentation transformations, including *random swapping*, *random insertion*, *random deletion*, and *synonym replacement*. In step 1, among the available strategies, a strategy is predicted using policy neural network. In step 2, the original dataset is augmented using the augmentation policy selected in step 1. In step 3, the training network is trained on the augmented dataset, with the accuracy serving as the evaluation metric. In step 4, the change in accuracy is utilized as the reward of the selected augmentation strategy to update and improve the policy network.

Finding the best data augmentation policy is formulated as an optimization problem. The policy network predicts the data augmentation policy (π). This policy contains information about the augmentation transformations and the probability of applying each transformation, which determine what percentage of the dataset should be augmented using which transformation. The policy π is employed to augment the original dataset, which is subsequently utilized to train a neural network with a fixed architecture referred to as the training network. Then, the accuracy (R) is used to calculate the reward for updating the policy network. The details of RLAug2Sent have been presented in the following.

3.2. Preprocessing

In NLP, it is necessary to preprocess the text to convert it into a suitable format for input. In this study, considering the dataset structure (unstructured text) and the desired objective, which is data augmentation, the following preprocessing steps have been applied to the dataset. We took advantages of scikit-learn library of python programming languages for performing tokenization, stemming, removing stop words, and embedding which was done using one-hot encoding.

3.3. Data Augmentation

For data augmentation, transformations need to be applied to the input data to generate new data. Generating an augmented image in computer vision is comparatively simpler. Even with the addition of noise or cropping a portion of the image, the model can still successfully classify the image. However, this does not hold true for NLP where data augmentation needs to be performed with caution due to the syntactic structure of the text. For example, not every word can be replaced with others like "a," "an," or "the." Additionally, not all words have synonyms. Even by changing a single word, the context will be completely different. In this paper, four data augmentation transformations at the *word level* are considered, including random deletion, random swapping, synonym replacement, and random insertion.

For implementing the four text augmentation transformations, EDA library of python language was used.

3.4. Augmentation Policy

In the proposed approach, the problem of identifying the optimal data augmentation policy is formulated as a discrete search problem. The search

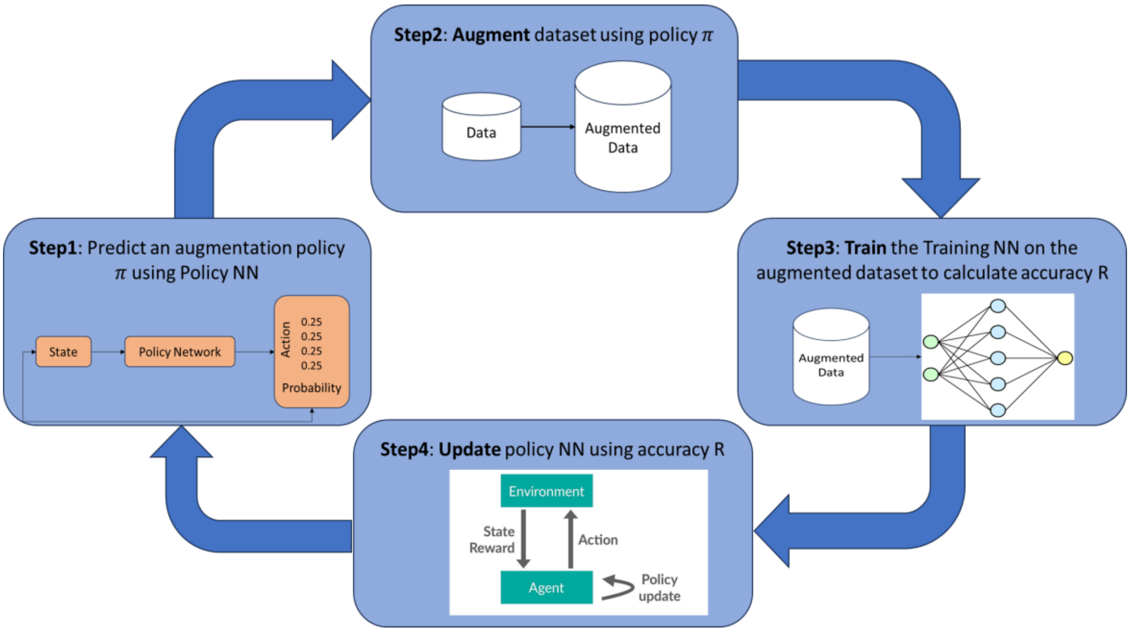

Figure. 1. Architecture of RLAug2Sent

algorithm (Policy network) which is implemented as a Deep-Q-Network or DQN, produces a data augmentation policy.

A policy comprises the probabilities assigned to text data augmentation transformations that are to be applied on the original dataset. Each augmentation policy is represented by a quadruplet, where each element represents the probability of a data augmentation transformation. In each iteration, based on the degree of change in the accuracy of the training network which is considered as the reinforcement learning reward, the probabilities of data augmentation transformations are adjusted, and a new policy is generated.

If the current policy leads to an increase in accuracy, it is considered a good policy. The state in reinforcement learning represents the current data augmentation strategy.

Therefore, the reinforcement learning components in the proposed method are defined as follows:

- **Policy**: Each policy determines the percentages of original dataset on which augmentation transformations should be applied. It specifies the percentage of data to be deleted, replaced, swapped, or inserted. The sum of these percentages equal to 100. For example, a policy can be defined as applying random deletion on 20%, synonym replacement on 40%, random swapping on 15%, and random insertion on 25% of the original datasets.
- **State**: Each state represents the current policy, a vector of length 4, with each element indicating the probability of the corresponding augmentation transformation.
- **Reward**: Following the augmentation of the original dataset based on the current data augmentation policy, the accuracy of training the network on the augmented dataset is calculated. If the accuracy has improved compared to the previous policy, the reward is positive; otherwise, the reward is negative.
- **Action**: Each action corresponds to changes in the current state (policy). More precisely, five actions have been defined, which is one more than the number of augmentation transformations. Four out of the five actions correspond to the four augmentation transformations, determining the transformation probability that should be increased by 3%, while the probabilities of the others should be decreased by 1%. The 5th action implies that no changes should be made to the current policy. For example, let's assume that the policy assigns a 25% share to the random insertion. Then it is decided to increase this percentage to 28%, resulting in a positive action of 3%. The action is applied to the policy, and a new policy is created. The percentages of the other three transformations are adjusted (decreased by 1%) to ensure that the total sum of all four percentages equals 100.

3.5. Search Algorithm

For policy network a DQN has been used. DQN combines Q-learning, a reinforcement learning approach, with deep neural networks [33]. Q-learning [34] is a model-free off-policy algorithm that operates iteratively and updates the action-value function according to the equation given in Equ. (1):

$$Q(s,a) \leftarrow (1-\alpha)\,Q(s,a) + \alpha\,[r(s,a) + \gamma \max(Q(s',a'))] \tag{1}$$

Where, α represents the learning rate, $r(s,a)$ is the most recent reward (in this case, the evaluated accuracy returned by the training network), and $\max(Q(s',a'))$ is the maximum Q-value among all possible actions in the next state, denoted by s'. This update policy essentially combines the current estimate of the Q-value for the state-action pair with a new estimate derived from the observed immediate reward and the maximum expected future reward. As the agent explores the environment and gains experience over time, the Q-values are updated iteratively through this policy, eventually converging to the optimal Q-values, which indicate the expected cumulative rewards for every combination of action and state in the environment.

$Q(s,a)$ denotes the expected cumulative reward associated with taking action a in a particular state s. The selection probability of action a is influenced by the policy, which outlines the likelihood of opting for each possible action within a specified state.

The key advantage of DQN over tabular Q-learning is its ability to handle high-dimensional state spaces, generalize to similar states, and automatically extract features. The update rule for DQN is akin to that of Q-learning, except that it approximates $Q(s,a)$ using a deep neural network.

3.6. Neural Networks Architectures and Hyperparameters

The policy network (DQN) is a three-layer fully connected neural network with 24 hidden units at each layer and ReLU activation functions. The input size is four, representing the number of augmentation transformations. Additionally, the output layer has a linear activation function, and its size is equal to 5. Four out of the five actions correspond to the four augmentation transformations, determining the transformation probability that should be increased. The 5th action implies that no changes should be made to the current policy. In the hyperparameter settings of DQN, a discount factor of 0.97 and a learning rate of 0.05 have been selected, with a batch size of 24. Also, the loss function used was the mean squared error (MSE).

For the training network, various neural network models were employed, including CNN, Bi-LSTM, Transformer, BERT, and XLNet. In terms of neural network hyperparameters, the maximum number of input features was set to 2000, and the batch size was set to 128. The activation function utilized was the sigmoid function, and the loss function used was the categorical cross entropy.

4. Result and Discussion

In this section, the proposed method is evaluated empirically. First, an explanation about the dataset is presented. Then, the evaluation results of RLAug2Sent with each of the neural networks as training network, including CNN [39], Bi-LSTM [40], Transformer [41], BERT [42], and XLNet [43], are described and compared with the previous methods.

4.1. Dataset

For the evaluation of this research, the English SemEval 2017 Task 4A-4C datasets [34], which focuses on sentiment analysis of Twitter, is used. In Figure 2**Error! Reference source not found.**, the dataset schema is represented. Also, the dataset characteristics are depicted in Table 1.

4.2. Comparison results with Data Boost using different training networks

In Table 2, the evaluation results based on the accuracy metric (which were considered as the evaluation metric in the previous work) are presented for the proposed method compared to the baseline paper [10]. RLAug2Sent is compared to two other methods in the table, including the primary neural network architecture (without data augmentation) and the Data Boost method.

The data is divided into three parts, where in the first run, 20% of the original data is used (plus the same amount of data generated using the data augmentation method), in the second run, 40% of the original data is used (plus the same amount of data generated by data augmentation method), and in the third run, 80% of the original data is separated, and the proposed method is executed on them and compared to Data Boost. The test data is the same for all methods.

The results show that the proposed method improves accuracy when using 20% and 40% data. Also, the proposed method outperforms Data Boost in both 20% and 40% of data.

As seen in Table 2, the XLNet network has the best results among various training networks, and had the highest accuracy. In the proposed method, it is also observed that using the XLNet network increases accuracy and is the best model for training.

	sentiment	text
0	neutral	Picturehouse's, Pink Floyd's, 'Roger Waters: T...
1	neutral	Order Go Set a Watchman in store or through ou...
2	negative	If these runway renovations at the airport pre...
3	neutral	If you could ask an onstage interview question...
4	positive	A portion of book sales from our Harper Lee/Go...

Figure. 2. Dataset SemEval 2017 Task 4A

Table 1. SemEval 2017 Task4A-C dataset characteristics

Dataset	*# Class*	*Train*	*Test*	*# Train Samples*	*# Test Samples*
SemEval 2017 Task 4A	3	39.64%	19.33%	49570	12284
		44.78%	48.33%		
		15.58%	32.33%		
SemEval 2017 Task 4B	2	78.85%	39.82%	18894	6185
		21.15%	60.18%		
SemEval 2017 Task 4C	5	3.34%	1.06%	30432	12379
		42.23%	18.84%		
		42.35%	50.04%		
		11.11%	28.64%		
		0.97%	1.43%		

Table 2. The evaluation results using various neural networks

Classifier	*Sentiment Analysis*		
	*20% (*2)*	*40% (*2)*	*80%*
CNN	0.458	0.502	0.557
CNN+Data Boost	0.477	0.527	-
CNN+ RLAug2Sent	0.509	0.529	-
Bi-LSTM	0.439	0.515	0.564
Bi-LSTM+Data Boost	0.513	0.542	-
Bi-LSTM+ RLAug2Sent	0.516	0.535	-
Transformer	0.371	0.458	0.551
Transformer+Data Boost	0.502	0.521	-
Transformer+ RLAug2Sent	0.504	0.523	-
BERT	0.514	0.582	0.679
BERT+Data Boost	0.610	0.642	-
BERT+ RLAug2Sent	0.621	0.645	-
XLNet	0.624	0.643	0.697
XLNet+Data Boost	0.636	0.657	
XLNet+RLAug2Sent	**0.651**	**0.693**	

The results also indicate that by utilizing only 20% and 40% of the original dataset, we can attain accuracy levels comparable to those achieved with 80% of the dataset (training set).

4.3. Comparison with previous data augmentation methods

In Table 3, the results of the proposed method are compared with other previous text data augmentation methods based on the F-Score evaluation metric (based on reported results in [10]) for 10% and 40% of the data. As can be seen, the proposed method has achieved better results compared to previous methods in terms of F1-score metric. The closest method to the proposed approach is Data Boost, which also demonstrates the effectiveness of using reinforcement learning-based methods for data augmentation.

Table 3. Comparison of evaluation results with previous data augmentation methods

Method	*Ref.*	*Sentiment Analysis*	
		10%	*40%*
Naïve Aug.	[23]	0.566	0.609
Word Replace Aug.	[36]	0.585	0.606
EDA	[6]	0.560	0.608
Word2Vec Aug	[37]	0.557	0.619
Contextual Word Embs Aug.	[5]	0.610	0.627
Back-Translation Aug.	[38]	0.617	0.620
Data Boost	[10]	0.591	0.642
RLAug2Sent	-	**0.621**	**0.691**

5. Conclusions

In this paper, we proposed a deep reinforcement learning based algorithm for text data augmentation. The DQN method was utilized for data augmentation in the context of sentiment analysis. This method represented a fusion of reinforcement learning algorithms and deep learning techniques. Various deep learning networks, such as CNN, Bi-LSTM, Transformer, BERT, and XLNet, were employed in the study for the training. The evaluation was carried out using the SemEval dataset. Experimental results show that reinforcement learning can be used as an efficient approach for augmenting text datasets to achieve better results on sentiment analysis task.

One recent application of RL is to improve the search process in optimization problems, where the solution space is complex and large. The focus of these studies is on finding the appropriate coefficients and importance for each of the text transformation functions under examination. Due to the continuous nature of these coefficients, the solution space is large, and conventional methods struggle to find optimal or near-optimal solutions. The advantage of using RL in such problems lies in leveraging the data generated in the search process so far, to enhance the subsequent search process. In this type of RL application, the state represents the current values of decision variables, and the action denotes the degree of change in these values. The values that these actions obtain in different states during the search process guide us towards the optimal solution.

There are various avenues for future research that can further build upon the findings and contributions of this study. Because of limited resources we could not compare our results with more previous methods and on more datasets. In the future, we intend to conduct a more comprehensive evaluation of the proposed approach.

In this paper, four transformations have been employed, which could, in some instances, change sentiment of a sentence, leading to a different label for the new sample compared to the previous one. Indeed, this is one of the fundamental challenges in text data augmentation compared to the image data augmentation. The approach taken in this paper to mitigate this challenge is through a reward mechanism, where the transformation that induces less sentiment alteration receives a higher reward and is utilized more frequently. It is emphasized that this approach does not solve the mentioned problem but rather adjusts it. This issue can be considered as another future research direction.

As a search mechanism, we only evaluated a single RL method, DQN. For future work, other RL methods, especially actor-critic methods, can be considered, as they may potentially enhance the results.

Also, we only considered a single NLP task, sentiment analysis. One can apply a similar method to other well-known tasks, such as fake news detection, stance detection, and opinion mining.

Declarations

Funding

This research did not receive any grant from funding agencies in the public, commercial, or non-profit sectors.

Authors' contributions

MR: All phases including study design, acquisition of data, interpretation of the results, statistical analysis, drafting and revision of the manuscript were caried out with Mehdy Roayaei

Conflict of interest

The authors declare that no conflicts of interest exist.

References

[1] R. Raileanu, M. Goldstein, D. Yarats, I. Kostrikov and R. Fergus, “Automatic Data Augmentation for Generalization in Deep Reinforcement Learning,” *arXiv preprint arXiv:2006.12862*, 2020, https://doi.org/10.48550/arXiv.2006.12862

[2] X. Wang, K. Wang and S. Lian, "A survey on face data augmentation for the training of deep neural networks," *Neural Comput Appl*, vol. 32, no. 19, pp. 15503–15531, 2020, http://doi.org/10.1007/s00521-020-04748-3.

[3] J. Ratner, H. R. Ehrenberg, Z. Hussain, J. Dunnmon and C. Ré, "Learning to compose domain-specific transformations for data augmentation," *arXiv*. 2017, https://doi.org/10.48550/arXiv.1709.01643.

[4] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J Big Data*, vol. 6, no. 1, 2019, http://doi.org/10.1186/s40537-019-0197-0.

[5] B. Li, Y. Hou, and W. Che, "Data augmentation approaches in natural language processing: A survey," *AI Open*, vol. 3, pp. 71–90, 2022, http://doi.org/10.1016/j.aiopen.2022.03.001.

[6] S. Y. Feng *et al.*, "A Survey of Data Augmentation Approaches for NLP," *arXiv preprint arXiv:2105.03075*., 2021, https://doi.org/10.48550/arXiv.2105.03075.

[7] S. Kobayashi, "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*, 2018, pp. 452-457, http://doi.org/10.18653/v1/N18-2072.

[8] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 6382–6388. http://doi.org/10.18653/v1/d19-1670.

[9] L. F. A. O. Pellicer, T. M. Ferreira and A. H. R. Costa, "Data augmentation techniques in natural language processing," *Appl Soft Comput*, vol. 132, p. 109803, Jan. 2023, http://doi.org/10.1016/j.asoc.2022.109803.

[10] C. Shorten, T. M. Khoshgoftaar and B. Furht, "Text Data Augmentation for Deep Learning," *J Big Data*, vol. 8, no. 1, p. 101, Dec. 2021, http://doi.org/10.1186/s40537-021-00492-0.

[11] R. Liu, G. Xu, C. Jia, W. Ma, L. Wang and S. Vosoughi, "Data Boost: Text Data Augmentation Through Reinforcement Learning Guided Conditional Generation," Proceedings of the 2020 Conference on Empirical *Methods in Natural Language Processing (EMNLP)*, 2020, pp. 9031–9041. http://doi.org/10.18653/v1/2020.emnlp-main.726.

[12] G. Raille, S. Djambazovska, & C. Musat, "Fast cross-domain data augmentation through neural sentence editing," *arXiv preprint arXiv:2003.10254*.

https://doi.org/10.48550/arXiv.2003.10254

[13] R. Hataya, J. Zdenek, K. Yoshizoe and H. Nakayama, "Faster AutoAugment: Learning Augmentation Strategies Using Backpropagation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12370 LNCS, pp. 1–16, 2020, http://doi.org/10.1007/978-3-030-58595-2_1.

[14] G. Daval-Frerot and Y. Weis, "WMD at SemEval-2020 Tasks 7 and 11: Assessing humor and propaganda using Unsupervised Data Augmentation," in *14th International Workshops on Semantic Evaluation, SemEval 2020 - co-located 28th International Conference on Computational Linguistics, COLING 2020, Proceedings*, 2020, pp. 1865–1874. http://doi.org/10.18653/v1/2020.semeval-1.246.

[15] T. Dao, A. Gu, A. J. Ratner, V. Smith, C. De Sa and C. Ré, "A kernel theory of modern data augmentation," in *36th International Conference on Machine Learning, ICML 2019*, 2019, pp. 1528–1537.

[16] D. Zhang, T. Li, H. Zhang, and B. Yin, "On Data Augmentation for Extreme Multi-label Classification," *arXiv preprint arXiv:2009.10778.*, 2020, https://doi.org/10.48550/arXiv.2009.10778

[17] X. Zuo, Y. Chen, K. Liu and J. Zhao, "KnowDis: Knowledge Enhanced Data Augmentation for Event Causality Detection via Distant Supervision," in *COLING 2020 - 28th International Conference on Computational Linguistics, Proceedings of the Conference*, 2020, pp. 1544–1550, http://doi.org/10.18653/v1/2020.coling-main.135.

[18] X. Dai and H. Adel, "An Analysis of Simple Data Augmentation for Named Entity Recognition," *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 3861–3867, http://doi.org/10.18653/v1/2020.coling-main.343.

[19] S. Longpre, Y. Wang and C. DuBois, "How effective is task-agnostic data augmentation for pretrained transformers?," in *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*, 2020, pp. 4401–4411. http://doi.org/10.18653/v1/2020.findings-emnlp.394.

[20] C. Rastogi, N. Mofid and F.-I. Hsiao, "Can We Achieve More with Less? Exploring Data Augmentation for Toxic Comment Classification," *arXiv preprint arXiv:2007.00875*., 2020, https://doi.org/10.48550/arXiv.2007.00875.

[21] B. Peng, C. Zhu, M. Zeng and J. Gao, "Data Augmentation for Spoken Language Understanding via Pretrained Models," *arXiv preprint arXiv:2004.13952*., 2020, https://doi.org/10.48550/arXiv.2004.13952.

[22] G. Yan, Y. Li, S. Zhang, and Z. Chen, "Data Augmentation for Deep Learning of Judgment Documents," In *Intelligence Science and Big Data Engineering. Big Data and Machine Learning: 9th International Conference, IScIDE 2019*, Nanjing, China, October 17–20, 2019, pp. 232–242. https://doi.org/10.1007/978-3-030-36204-1_19.

[23] C. Coulombe, "Text Data Augmentation Made Simple By Leveraging NLP Cloud APIs," 2018, [Online]. Available: http://arxiv.org/abs/1812.04718

[24] M. Regina, M. Meyer and S. Goutal, "Text Data Augmentation: Towards better detection of spear-phishing emails," *arXiv preprint arXiv:2007.02033*., 2020, https://doi.org/10.48550/arXiv.2007.02033

[25] J. Min, R. Thomas McCoy, D. Das, E. Pitler and T. Linzen, "Syntactic data augmentation increases robustness to inference heuristics," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2339–2352, https://doi.org/10.18653/v1/2020.acl-main.212.

[26] Y. Zhang, T. Ge and X. Sun, "Parallel data augmentation for formality style transfer," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 3221–3228, https://doi.org/.18653/v1/2020.acl-main.294.

[27] A. Anaby-Tavor *et al.*, "Do not have enough data? Deep learning to the rescue!," in *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, 2020, pp. 7383–7390, https://doi.org/10.1609/aaai.v34i05.6233.

[28] N. Thakur, N. Reimers, J. Daxenberger and I. Gurevych, "Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 296–310, https://doi.org/10.18653/v1/2021.naacl-main.28.

[29] H. Guo, Y. Mao and R. Zhang, "Augmenting Data with Mixup for Sentence Classification: An Empirical Study," *arXiv preprint arXiv:1905.08941*., 2019, https://doi.org/10.48550/arXiv.1905.08941.

[30] P. Yu, R. Zhang, Y. Zhao, Y. Zhang, C. Li and C. Chen, "SDA: Improving Text Generation with Self Data

Augmentation," *arXiv preprint arXiv:2101.03236*., 2021, https://doi.org/10.48550/arXiv.2101.03236.

[31] J. Fang and P. Li, "Data Augmentation with Reinforcement Learning for Document-Level Event Coreference Resolution," In *Natural Language Processing and Chinese Computing: 9th CCF International Conference, NLPCC 2020*, Zhengzhou, China, 2020, pp. 751–763, https://doi.org/10.1007/978-3-030-60450-9_59.

[32] J. Kim and K. E. Kim, "Data Augmentation for Learning to Play in Text-Based Games," in *Proc. IJCAI*, 2022, pp. 3143-3149, https://doi.org/10.24963/ijcai.2022/436.

[33] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015, https://doi.org/10.1038/nature14236.

[34] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3–4, pp. 279–292, May 1992, https://doi.org/10.1007/BF00992698.

[35] "SemEval 2017 Task 4A." [Online]. Available: https://alt.qcri.org/semeval2017/task4/

[36] T. Niu and M. Bansal, "Adversarial Over-Sensitivity and Over-Stability Strategies for Dialogue Models," *arXiv preprint arXiv:1809.02079.*, 2018, https://doi.org/10.48550/arXiv.1809.02079.

[37] W. Y. Wang and D. Yang, "That's So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets," *In Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 2557-2563, https://doi.org/10.18653/v1/D15-1306.

[38] W. Yu *et al.*, "QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension," *arXiv preprint arXiv:1804.09541*., Apr. 2018, https://doi.org/10.48550/arXiv.1804.09541.

[39] H. Wang, J. He, X. Zhang and S. Liu, "A short text classification method based on N-gram and CNN," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 248-254, 2020, https://doi.org/10.1049/cje.2020.01.001.

[40] B. Jang, M. Kim, G. Harerimana, S. U. Kang and J. W. Kim, "Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism," *Applied Sciences*, vol. 10, no. 17, pp. 5841, 2020, https://doi.org/10.3390/app10175841.

[41] Singh, Sushant, and Ausif Mahmood. "The NLP cookbook: modern recipes for transformer based deep learning architectures," *IEEE Access*, vol. 9, pp. 68675-68702, 2021, https://doi.org/10.1109/ACCESS.2021.3077350..

[42] H. Li, Y. Ma, Z. Ma and H. Zhu, "Weibo text sentiment analysis based on BERT and deep learning," *Applied Sciences*, vol. 11, no. 22, pp. 10774, 2021, https://doi.org/10.3390/app112210774.

[43] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

Mehdy Roayaei received his B.S., M.S., and Ph.D. in Computer Engineering from Amirkabir University of Technology (AUT) in 2008, 2010, 2016. He is currently an Assistant Professor of Computer Engineering at Tarbiat Modares University. He is interested in using reinforcement learning approaches for handling complex problems in real-world environments.