Received: 11 August 2020 | Revised: 21 April 2021 | Accepted: 29 April 2021

DOI: 10.1002/asi.24493

RESEARCH ARTICLE

JASIST WILEY

Lexical data augmentation for sentiment analysis

Rong Xiang[1] | **Emmanuele Chersoni**[1] | **Qin Lu**[1] | **Chu-Ren Huang**[1] | **Wenjie Li**[1] | **Yunfei Long**[2]

[1]The Hong Kong Polytechnic University, Hong Kong SAR, China

[2]University of Essex, Essex, UK

Correspondence
Yunfei Long, University of Essex, School of Computer Science and Electronic Engineering, Essex, UK.
Email: yl20051@essex.ac.uk

Abstract

Machine learning methods, especially deep learning models, have achieved impressive performance in various natural language processing tasks including sentiment analysis. However, deep learning models are more demanding for training data. Data augmentation techniques are widely used to generate new instances based on modifications to existing data or relying on external knowledge bases to address annotated data scarcity, which hinders the full potential of machine learning techniques. This paper presents our work using part-of-speech (POS) focused lexical substitution for data augmentation (PLSDA) to enhance the performance of machine learning algorithms in sentiment analysis. We exploit POS information to identify words to be replaced and investigate different augmentation strategies to find semantically related substitutions when generating new instances. The choice of POS tags as well as a variety of strategies such as semantic-based substitution methods and sampling methods are discussed in detail. Performance evaluation focuses on the comparison between PLSDA and two previous lexical substitution-based data augmentation methods, one of which is thesaurus-based, and the other is lexicon manipulation based. Our approach is tested on five English sentiment analysis benchmarks: SST-2, MR, IMDB, Twitter, and AirRecord. Hyperparameters such as the candidate similarity threshold and number of newly generated instances are optimized. Results show that six classifiers (SVM, LSTM, BiLSTM-AT, bidirectional encoder representations from transformers [BERT], XLNet, and RoBERTa) trained with PLSDA achieve accuracy improvement of more than 0.6% comparing to two previous lexical substitution methods averaged on five benchmarks. Introducing POS constraint and well-designed augmentation strategies can improve the reliability of lexical data augmentation methods. Consequently, PLSDA significantly improves the performance of sentiment analysis algorithms.

1 | INTRODUCTION

Sentiment analysis is a text mining technique that uses machine learning and natural language processing (NLP) methods to automatically analyze sentiment expressed in text (positive, negative, neutral, and beyond). In recent years, deep learning methods have achieved significant improvements for sentiment analysis over previous

machine learning methods (Long et al., 2019; Tang et al., 2015; Yang et al., 2016). Notably, transformer models like bidirectional encoder representations from transformers (BERT) (Devlin et al., 2019) and its refinement RoBERTa (Y. Liu et al., 2019), have achieved state-of-the-art performance in various sentiment analysis tasks. These deep learning approaches benefit from unsupervised representation learning to obtain unprecedented general language knowledge, using a large amount of unlabeled data (Devlin et al., 2019). Still, it is important to acquire task-specific or domain-specific knowledge from *data that are smaller than the training data needed to acquire general language knowledge* (Glorot et al., 2011). However, manually collecting domain specific training dataset is rather time-consuming and costly. To address the problem of labeled data scarcity, data augmentation is studied in this work to address the acquisition sufficient task-specific training data in the deep learning age.

Data augmentation is widely used in computer vision (Huang et al., 2017; Krizhevsky et al., 2012; Uzunova et al., 2017; Zagoruyko & Komodakis, 2016; Zhao et al., 2019) to control generalization error for deep learning models. These augmentation techniques usually work by applying transformations (such as image shift, rotation, and shear mapping) to available instances, generating new image instances with minor semantic variations. In NLP, data augmentation was adopted as well when labeled data are expensive to acquire. However, text is written in a specific sequential order to convey specific semantic information subjected to certain syntactic constraints. Thus, many augmentation methods used in image processing may not be suited for NLP tasks due to the inherent complexity of natural language. Although it is generally difficult to devise a universal scheme, different augmentation methods have been proposed for different NLP tasks, such as machine translation (Sennrich et al., 2015), dialogue systems (Kurata et al., 2016), question answering (Fader et al., 2013) and in classification tasks including sentiment analysis (R. Xu et al., 2015; X. Zhang et al., 2015; Z. Hu et al., 2017; Kobayashi, 2018; Kishimoto et al., 2018; Wei & Zou, 2019; S. Liu et al., 2020).

Lexical substitution is a natural and effective approach in the augmentation of text data (Wei & Zou, 2019). This approach typically selects words of the original sentence with replacement of their semantic neighbors. An early lexical substitution method made use of a thesaurus as a resource to replace words with their synonyms (X. Zhang et al., 2015). WordNet (Miller, 1995) is another commonly used resource for synonym replacement (Wei & Zou, 2019). In addition to using well-structured knowledge resources, interpolation by word embedding is also a feasible way to generate semantically close candidates for substitution (W. Y. Wang & Yang, 2015; R. Xu et al., 2015). In these works, however, word replacement methods and augmentation strategy have not been further explored.

To investigate the effectiveness of lexical data augmentation for machine learning models, a new method is proposed in this work to use part-of-speech (POS) focused lexical substitution for data augmentation (PLSDA). As the purpose of PLSDA is to create labeled training data in the form of natural language text, lexical substitution must consider both syntactic correctness and semantic closeness. More specifically, PLSDA first makes use of POS tags to determine words to be replaced for syntactic consistency. Semantic criteria are then used to obtain replacements by taking into account of both similarity and diversity. The main contributions of this work are listed as follows:

- *A new data augmentation method*: A new PLSDA approach is proposed, which uses PLSDA to further improve performance. Performance evaluations demonstrate that augmentation using PLSDA leads to about 1.4% of accuracy gain compared to training of none-augmented datasets across all the classifiers and benchmarks. The improvement is also significantly higher compared to previous lexical substitution-based augmentation methods. In addition, an in-depth parameter study on PLSDA is conducted to optimize the model performance.
- *Detailed efficiency analysis of different replacement types*: Investigations show that nouns and adjectives/adverbs are better POS types for the proposed PLSDA method, while verbs have minimal benefit in the substitution process. In addition to synonyms, hypernyms (HPE) are also found to be good replacements for data augmentation. Notably, they are useful when the synonym set is insufficient. Furthermore, the importance of the diversity of augmented instances is discussed by comparing two different substitution strategies.

The rest of this article is organized as follows. Section 2 introduces related works in data augmentation. Section 3 describes detailed design of the proposed PLSDA approach. Section 4 gives details on lexical data augmentation methods, sentiment classifiers and benchmark datasets. Detailed results and performance analysis are presented in Section 5. Finally, Section 6 gives concluding remarks and future direction.

2 | RELATED WORK

To address the scarcity problem of labeled data in NLP tasks, Fadaee et al. (2017) presented a novel data augmentation approach in machine translation that targets low-frequency words by generating new sentence pairs

containing rare words in new, synthetically created contexts. Later on, another method in machine translation used the so-called SwitchOut method (X. Wang et al., 2018) to replace words in the source and the target sentences with random words from the vocabulary. In text classification, Y. Xu et al. (2016) applied data augmentation in relation classification by changing the order of subpaths in the dependency tree to obtain reversed relationships. Their approach leads to an improvement of 1.9% compared to models using the original training set. For dialogue language understanding, Hou et al. (2018) made use of a sequence-to-sequence model to generate lexical and syntactic alternatives for a given utterance and leveraged them to augment the training data, achieving a significant improvement on two testing datasets. Niu and Bansal (2018)) presented two categories of model-agnostic adversarial strategies that reveal the weaknesses of several generative, task-oriented dialogue models: Should-Not-Change strategies that evaluate oversensitivity to small and semantics-preserving edits, as well as Should-Change strategies that test if a model is overstable against subtle yet semantics-changing modifications. Another data augmentation approach attempted to invert the source sentence, reversing its meaning, to generate examples of the opposing class. That method has been proven useful for binary classification tasks (Tarasov, 2020).

Even though data augmentation has been used in different NLP tasks, it is relatively novel in sentiment analysis. Based on *word embedding vectors*, R. Xu et al. (2015) generated more minority class samples by interpolating two-word embedding vector instances of the same label to obtain balanced data. Although the method showed significant improvement, the newly created vector instances have no corresponding readable text. This model shows no interpretability as it is purely based on vector manipulation, and thus cannot be reverted to human-readable text. Another approach using word embedding-based augmentation on Tweets explored the cosine similarity of words and framed word representations as the metric to find a replacement for target words among their k-nearest-neighbors (W. Y. Wang & Yang, 2015). Despite the loss of some syntactic information, they obtained better performance on a topic classification task. Another work (Vijayaraghavan et al., 2016) on tweets stance detection employed Word2Vec (Mikolov et al., 2013) to find candidates those are ranked on the basis of cosine similarity between Word2Vec vectors. They also set up a threshold, which is 0.5, for cosine similarity to select a replacement. Focusing on lexicon augmentation, Xiang et al. demonstrated that syntactic restriction can be further applied to refine the selection of candidates (Xiang et al., 2020). Their system achieved 1.3% accuracy improvement on eight text classification benchmarks. In addition, Rizos et al. proposed three text-based data augmentation techniques (substitution-based augmentation, word position augmentation, neural generative augmentation), aimed at reducing the degree of class imbalance and to maximize the amount of information that can extract from limited resources (Rizos et al., 2019).

Lexical replacement via thesaurus or ontology is another popular strategy for lexical data augmentation. An approach proposed by X. Zhang et al. (2015) replaced lexical items for character-level convolutional neural networks (CNNs). They adopted two geometric distributions to determine the number of words to be substituted, and they chose the replacement from a ranked candidate list. They showed that augmented data improved the performance of CNNs, achieving the best performance. Wei and Zou introduced the easy data augmentation (EDA) method using four manipulations including synonym replacement, random insertion, random swap, and random deletion (Wei & Zou, 2019). EDA showed an average 0.8% accuracy improvement in five benchmarks: SST-2 (Socher et al., 2013), Customer reviews (M. Hu & Liu, 2004), Subj (Pang & Lee, 2004), TREC (Li & Roth, 2002), and Pro-Con dataset (Ganapathibhotla & Liu, 2008).

Recently, some novel methods for data augmentation via language models or deep learning models have been proposed. Variational autoencoder and attribute discriminator have been combined (Z. Hu et al., 2017) to produce pseudoinstances. The use of paradigmatic relations between vectors can also provide a wider range of replacement. Kobayashi (2018) used a bidirectional language model (BiLM) as context-aware data augmentation. The probability of a target word is computed forward and backward based on the probability distribution of its context. Based on the output of BiLM, words for augmentation are selected via an annealed distribution, and a 0.6% accuracy gain was reported on five datasets.

In these research studies, several methods are proposed to manipulate the original sentence. However, little attention was given to explain which parts of a sentence should be changed and how to refine the replacing process. The lack of interpretability of the augmented data is another problem in most of these methods.

3 | DESIGN PRINCIPLES OF PLSDA

Lexical substitution refers to methods that create new instances from a given dataset by replacing several target

words with substitutes according to certain principles. PLSDA consists of two main parts: *substitution candidate selection* and *instance generation*. For a given sample, *substitution candidate selection* first follows its **syntactic consistency principle** and uses POS constraints to select candidate lexicons for substitution. It then follows the **semantic consistency principle** to identify lexical units via semantic relatedness for each target word in order to form a *substitution candidate lists (SCLs)*. In *instance generation*, whether a target word is replaced or not is determined by sampling from Bernoulli distribution of *SCLs* , to form the final *substitution collection (SC)*. Finally, substitutes in *SC* corresponding positions are sampled to generate augmented instances.

3.1 | Substitution candidate selection

Let I denote a training instance composed of n words, $I = \{w_1, w_2, w_i, ..., w_n\}$. For each **target word** w_i, its POS tag t_{w_i} can be readily obtained from available tools such as the Stanford NLP pipeline (Manning et al., 2014; Toutanova et al., 2003). The replaced words with the same POS tag, according to the **syntactic consistency principle** constraint, ensure the correct syntactical identity of generated text. Substitutions are allowed to act on only certain target word classes in I such that the newly created samples are likely to make sense. For example, if we replace function words such as "of," "in," and so forth with other function words, such as "on the table" becomes "in the table," it may cause semantically incorrectness. In this work, we choose to replace only content words, that is, nouns, adjectives, adverbs, verbs.

In *substitution candidate selection* step, the substitution candidates of each w_i are obtained. Two methods are devised to obtain candidates in PLSDA, following the **semantic consistency principle**. In both methods, candidates are selected from WordNet (Fellbaum, 2010). Ensuring identical POS guarantees that candidates are replaced only by words that will not change the syntax. For example, the verb "chair" (act as chairperson of or preside over [an organization, meeting, or public event]) could not be replaced with the noun "bench."

In the first method, let SCL_{w_i} denote the SCL for each w_i with m synonyms. SCL_{w_i} is obtained according to the following formula:

$$SCL_{w_i} = \left\{ c^1_{w_i}, c^2_{w_i}, ... c^m_{w_i} \;\middle|\; c^j_{w_i} \in Syn(w_i, j) \,\&\, t_{w_i} = t_{c^j_{w_i}} \right\} \quad (1)$$

where $c^j_{w_i}$ is the jth synonym for target word w_i. $Syn(w_i, j)$ refers to the synonym of w_i, where j is the membership subscript. Only w_i with at least one or more synonyms will be considered in instance generation ($m > 0$).

In the second method, substitutions are based on ontological relations, including HPE and hyponyms (HPO), as they are also considered semantically similar words. Let $Hpe(w_i, j)$ denote the jth HPE and $Hpo(w_i, j)$ denote the jth HPO of w_i. This method is potentially useful when there are not sufficient members in SCL_{w_i} with the synset-based method.

A similarity checker is designed to measure the closeness of a new instance to the original sample. Pretrained glove vectors (Pennington et al., 2014) are used to compute the cosine similarity between w_i and every member in SCL_{w_i}. As a system parameter, the *similarity threshold TH* is introduced to remove candidates $c^j_{w_i}$ with $Sim\left(w_i, c^j_{w_i}\right) < TH$. Only substitutes that are above the similarity threshold will remain in *SCLs*.

To better illustrate the process of *substitution candidates selection*, Figure 1 shows the steps involved to obtain *SCLs* for the sentence a great script brought down by lousy direction. Let the POS constraints be adjective and adverb for demonstration purpose. Three words are then potentially replaceable: great, down, and lousy.

To demonstrate the quality guaranteed by similarity scores, great, and lousy are exemplified in Table 1 with several handpicked lexicons. Generally, candidate lexicons are more semantically close with higher cosine similarity than those with lower similarity scores. In this case study, *TH* is set to 0.7 as an example. Based on WordNet synset information, we can obtain the corresponding *SCLs* has three candidate lists: $SCL(\texttt{great}) = \{awesome, terrific, perfect\}$, $SCL(\texttt{lousy}) = \{awful, horrible, poor\}$, $SCL(\texttt{down}) = \emptyset$. Note that the target word down here is an Adverb. Even though down, as an adjective, does have synonyms such as "gloomy" and "depressed," they will not appear in its *SCL*. The target word down serves as an example to show that with the POS constraint, some words may not have available *SCLs*. Consequently, they will not serve as replaceable target word when the synonym strategy is used. At this point, cosine similarity is calculated for each synonym. In this case, awesome is filtered out as its similarity is lower than the value of *TH*. These *SCLs*, obtained by PLSDA in the substitution candidate selection step, are now ready to produce augmented instances.

3.2 | Instance generation

Let k denote the number of positions in I that have qualified *SCLs* in a sample sentence and s be the average number of qualified candidates in each position. Then, the potential number of generated sentences is in the order

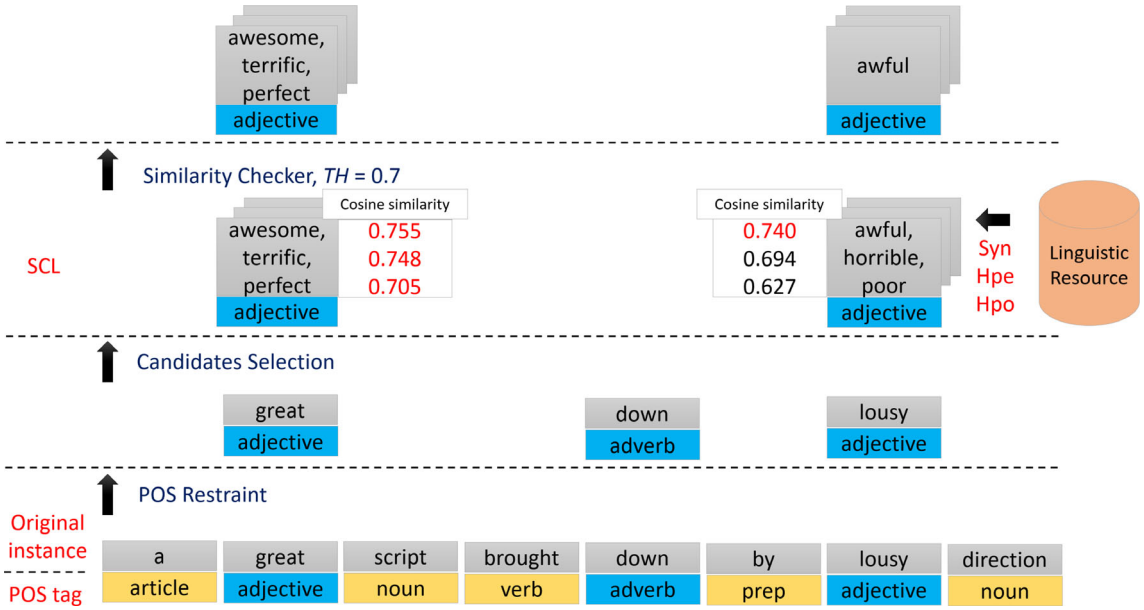

FIGURE 1 Example of substitution candidate selection using synonyms [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 1 Candidate examples in different similarity interval

	<0.4	<0.5	<0.6	<0.7	<0.8	<0.9
Great	Wonderfulness ingenuity	Nice wonderful	Good considerable	Tremendous immense	Perfect terrific	NA
Lousy	Hypocrite embarrassingly	Implausible suspicious	Annoying damn	Ridiculous silly	Awful crappy	NA

of $O(s^k)$, which means that if k is large, a large number of instances will be generated if there is no screening process. To control the size of generated instances, a two-step sampling approach is used to determine appropriate values of k and s. The first-step sampling determines w_i at which positions should be replaced. The second-step sampling selects candidates from the s substitutes.

To determine the position, sampling is conducted from a collection of Bernoulli distribution $Ber(p_s)$ for every SCL_{w_i} with probability p_s. When there is no prior-knowledge, $p_s = 0.5$ can be naively used, which means that each position has 50% chance of being selected. The Bernoulli distribution below determines whether w_i with a nonempty SCL is selected as replacement points to form the final SC.

$$P(w_i) = p_s^x (1-p_s)^{1-x} \begin{cases} x=1 \; w_i \; is \; selected, \; SC = SC \cup SCL_{w_i} \\ x=0 \; w_i \; is \; not \; selected. \end{cases} \tag{2}$$

This part can use different augmentation strategies to generate new instances efficiently. These strategies can be formulated in two directions.

The first strategy, referred to as the **stochastic strategy**, randomly picks a candidate in $SCLs$ to avoid a rigorous selection. This random process is designed as sampling from a Categorical distribution of POS tags $Cat\left(p_{w_i}^1, p_{w_i}^2, ..., p_{w_i}^j, ..., p_{w_i}^m\right)$, where $\sum p_{w_i}^j = 1$. Equivalent probability $p_{w_i}^j$ for each $c_{w_i}^j$ is used in our implementation.

$$P\left(X = c_{w_i}^j\right) = p_{w_i}^j, \; j \in [1, m] \tag{3}$$

The second strategy, referred to as the **similarity-first strategy**, uses similarity measures to pick the candidates, exploiting similarity ranking. To use the similarity-first strategy, candidates $\left\{c_{w_i}^1, c_{w_i}^2, ...c_{w_i}^m\right\}$ for a target word w_i need be sorted first according to their cosine similarity of word vectors. Augmented instances are then generated following the position of the candidates in the ranking.

Figure 2 shows an illustration of instance generation using the stochastic strategy for the same sample given in Figure 1. Starting from $SCLs$, great and lousy in the original sentence have multiple substitute options. Sampling from Bernoulli, SCL of great is chosen as the SC. Then, terrific is selected by categorical distribution and thus, a

new instance can be generated. This process can be executed repeatedly until sufficient instances are generated.

Since the complexity for finding substitutions is in the order of $O(s^k)$, it is not sufficient to select all possible replacements. The number of generated samples referred to *expected generated instances (EGI)*, is introduced in the *sampling* step as an algorithm termination parameter. This parameter will be determined empirically. Instance generation is done iteratively to create up to *EGI* number of new instances. Any generated instance shall be recorded to prevent it from being generated again.

In most cases, the algorithm will terminate when the iteration reaches *EGI*. If I is a concise sentence, it may have a minimal number of *SCLs* and replacement options. In general, there are cases in which the number of generated instances will not reach *EGI*.

3.3 | Instance generation examples

When generating instances, the utmost importance is to assure the identity of sentiment label between the generated and original instances. Four generated instance examples are presented to demonstrate how new instances are generated. These sentences are selected from sentiment analysis dataset SST-2, augmented with PLSDA using the **stochastic strategy** as the exemplary case. The original instances are given in the first line in *italic* and the target words to be substituted are marked by underscores. The corresponding replacement words are highlighted in **bold**.

Example 1 has a negative label. "language," "dreary," and "sluggish" are three target words for replacement. Their corresponding *SCLs* are 5, 10, and 6, respectively. Potentially, there are 461 different replacement combinations. In order to control the number of generated instances for one training sentence, *EGI* is set to five.

Example 1, label: negative

- *Without Shakespeare's eloquent language, the update is dreary and sluggish.*
- Without Shakespeare's eloquent **speech**, the update is dreary and sluggish.
- Without Shakespeare's eloquent **speech**, the update is dreary and **dull**.
- Without Shakespeare's eloquent **terminology**, the update is **drab** and sluggish.
- Without Shakespeare's eloquent **speech**, the update is dreary and sluggish.
- Without Shakespeare's eloquent language, the update is dreary and **dull**.

From the five generated instances we can see that the efficient stochastic method can still generate instances semantically close to the original training data and their inherited labels being negative are still correct.

Example 2 has a positive label. Two adjectives ("funny" and "beast") and two nouns ("horror" and "films") in the original sentence are selected to find replacement. The sizes of their respective *SCLs* are 17, 7, 3, and 9 for "funny," "beast," "horror," and "films," potentially generating 5,759 different instances. The five instances generated by the stochastic method are given below.

Example 2, label: positive

- *A stirring, funny and transporting re-imagining of beauty and the beast and 1930s horror films.*
- A stirring, funny and transporting re-imagining of beauty and the beast and 1930s **repugnance** films.

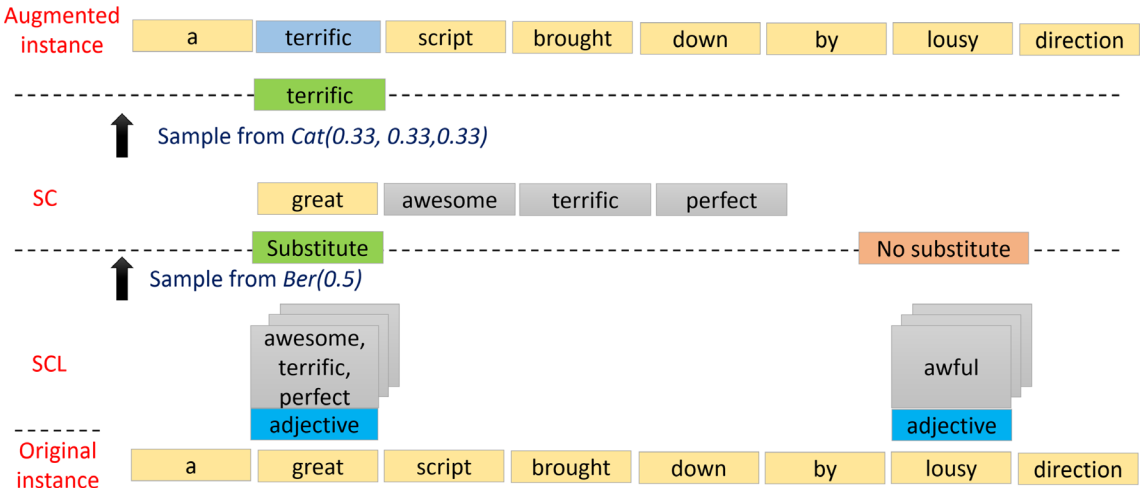

FIGURE 2 Example of instance generation [Color figure can be viewed at wileyonlinelibrary.com]

- A stirring, **amusing** and transporting re-imagining of beauty and the beast and 1930s **repugnance movie**.
- A stirring, funny and transporting re-imagining of beauty and the **animal** and 1930s horror **movie**.
- A stirring, funny and transporting re-imagining of beauty and the beast and 1930s horror **movie**.
- A stirring, **amusing** and transporting re-imagining of beauty and the **animal** and 1930s horror films.

The generated instances show that the meaning of the augmented samples is similar to that of the original one. Although the substituted word "animal" changes the film name "beauty and the beast" in some augmented instances, the sentiment label is still positive. This example indicates an issue with the noun phrases and proper noun compound in which the propose method does not conduct any further process. Semantic changes of this kind can be tolerated during the augmentation process because our primary goal is only to make sure that the newly generated instance can inherit the original label as well as a grammatically correct sentence.

Example 3 has a negative label. In this example, only two words, "same" and "old" are used as the target words. Their *SCLs* sizes are only 1 and 6. When using the stochastic method, PLSDA only produced three augmented instances as shown below.

Example 3, label: negative

- *Sayles ... once again strands his superb performers in the same old story.*
- Sayles ... once again strands his superb performers inthe **like erstwhile** story.
- Sayles ... once again strands his superb performers in the **like** old story.
- Sayles ... once again strands his superb performers in the same **erstwhile** story.

This example shows that sometimes, the number of augmented instances does not reach five, the *EGI* target. This is why HPE may be useful to enlarge the substitution vocabulary. Another issue in this example is that when "old" is replaced by "erstwhile," it breaks the idiom "the same old story," and the expression "erstwhile story" hardly conveys negative sentiment. This shows that, on some occasions, lexicon augmentation may introduce noisy training instances.

Example 4 is another negative example in which the sizes of *SCLs* are 1, 1, and 5 for "other," "ordinary," and "fashion." Under this condition, augmentation method may produce very few instances. Three randomly generated instances are listed below.

Example 4, label: negative

- *borrows from other movies like it in the most ordinary and obvious fashion.*
- borrows from other movies like it in the most **average** and obvious fashion.
- borrows from **early** movie like it in the most **average** and obvious fashion.
- borrows from other movies like it in the most **average** and obvious **manner**.

Semantically, replacing "other" with "early" also makes it challenging to identify the sentiment clue. The slight change of wording can lead to some minor sentiment change between the original sentence and the augmented one. However, it hardly changes the polarity.

These examples show that using lexical substitution can sometimes change the meaning of the original sentences. However, the augmented instances are less likely to have a significant change in sentiment. Despite a few inappropriate augmentation instances, most augmented instances are reliable. Anyhow, the slight shift in the degree of polarity can be tolerated as polarity takes only discrete value. In fact, these examples also indicate that PLSDA can introduce more diversity of training sentences, yet at the same time preserve sentiment polarity.

4 | METHODS AND DATASETS FOR EVALUATION

Three lexical data augmentation methods are investigated. Their implementation details and parameters are provided in Section 4.1. Six machine learning models are used as sentiment classifiers in the evaluation. Their settings are fine-tuned and given in Section 4.2. Five benchmark datasets for sentiment analysis and corresponding statistics are introduced in Section 4.3.

4.1 | Data augmentation methods

In general, all lexical data augmentation methods involve two main phases. The first phase selects appropriate words in an original instance so as to find their replacement and the second phase actually finds appropriate replacement words to generate new instances. In addition to PLSDA proposed in this work, two lexical data argumentation methods in literature are included in the evaluation for comparison with the following settings:

- **DICT** ass proposed by X. Zhang et al. (2015). They experimented data augmentation by using an English

thesaurus. The number of words r to be replaced is determined by a geometric distribution with parameter p, where $p[r] \sim p^r$. The index s of the synonym chosen given a word is also determined by another geometric distribution in which $p[s] \sim q^s$. They chose $p = 0.5$ and $q = 0.5$ in their implementation. On the basis of the distribution design, the number of replaced words is unlikely to be very large and the probability to find a synonym becomes small when it is not a commonly used meaning. The expected number of augmented sentences to generate per original sentence was not mentioned in their paper. Experimentally, we implemented their method with generated instance count set to 5 for fair comparison.

- **EDA** is a recent method introduced by Wei and Zou (2019). It consists of four simple lexicon operations: synonym replacement, random insertion, random swap, and random deletion. We follow their default settings in which up to 10% words can be changed randomly and five augmented sentences will be generated.
- **PLSDA** proposed in this work, applies the POS tag constraint. In the phase of *substitution candidate selection*, the best performances are achieved by fine-tuning the use of POS types (using noun and adjective/adverb and dropping verb), semantic augmentation set (synonyms, which outperforms HPE and HPO). The similarity threshold *TH* is tuned to 0.6. As for *instance generation*, whether each word shall be replaced or not depends on a Bernoulli distribution with .5 probability. To determine the substitution, stochastic augmentation strategy is used to randomly select a possible word with a POS-equivalent one via categorical distribution. The comparison between using similarity-first strategy is also investigated.

4.2 | Sentiment classifiers

To assess the contribution of data augmentation using PLSDA, six commonly used classifiers for sentiment analysis tasks are used in the evaluation. One is a traditional machine learning model (SVM) and five are deep learning models. SVM (Mullen & Collier, 2004) serves only as baseline to represent traditional machine learning methods. LSTM models are commonly used in NLP tasks because of their ability to handle sequential data such as text. In the evaluation, two popular LSTM variants are used including the basic LSTM model (Hochreiter & Schmidhuber, 1997) and the more comprehensive BiLSTM-AT model (Y. Zhang et al., 2018). For both LSTM models, pretrained GloVe vectors (Pennington et al., 2014) were used for word embedding initialization. For BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), and RoBERTa (Y. Liu et al., 2019), we fine-tuned the pretrained models on each dataset, respectively.

Table 2 summarizes the detailed settings for the six models. For fair comparison, algorithm settings are kept as much identical as possible. For transformer-based methods, the recommended settings in (Devlin et al., 2019; Yang et al., 2019; Y. Liu et al., 2019) are used for learning rate. Since LSTM variants use different word embedding schema, their learning rates are experimentally optimized to 0.0005. The embedding dimensions are determined by the pretrained data. Learning rates, dropout rate, batch size, and epoch size are all determined experimentally based on the performance. In practice, the optimized settings LSTM-based methods are different from BERT variants, being tuned to obtain better performance in the evaluation.

4.3 | Datasets

The following five datasets, used in the evaluation, are commonly used benchmarks for sentiment analysis in English.

- **SST-2** is a collection of movie reviews. The reviews were labeled as *positive* or *negative* by Socher et al. (2013).
- **MR** is another collection of Movie reviews with one sentence per review. Classification involves detecting *positive/negative* reviews (Pang & Lee, 2005).
- **IMDB** is a collection of movie reviews from IMDB website annotated by Maas et al. (2011).[1] The text is relatively long. Reviews are labeled as either *positive* or *negative* and the data is fully balanced with a *positive/negative* ratio of 1:1.
- **Twitter** is a dataset collected from twitter, available for public download.[2] Tweets and comments on several topics, such as news, public events and daily life, are included in this corpus. This dataset is designed for a typical sentiment analysis task as the labels concern the sentiment polarities which are *positive/negative*.
- **AirRecord** is a collection of customer conversation messages from Twitter from six major American airlines. The texts were collected in February 2015 and are available for public download.[3] The sentiment polarity was manually annotated with three class labels: *positive*, *negative*, and *neutral*.

Table 3 shows the statistics of the five datasets, including their training set size (N_{train}), testing set size (N_{test}), average length of the instances (L_{avg}, word count), *SD* of length (L_δ, total vocabulary size (N_{voc}), number of class labels (C), and instance number of each class (C_p :

positive, C_n : negative, C_{neu} : neutral). Out of the five datasets, Twitter has the largest number of instances, and IMDB has the longest average length. AirRecord is the only three-class classification dataset.

In data preprocessing, Python scripts are first used to remove special characters such as tabs, URLs, redundant blanks and {′ ″} (which hinder word matching before finding substitution). StanfordNLP tools (Manning et al., 2014) are used to tokenize each instance in the corpora. The training/testing dataset split of SST-2 is given by the data supplier. For the rest of the benchmark datasets, 80%/20% of total instances are randomly split as the training/test set. Three independent splits with different random seed are conducted. Performance evaluation is reported based on the average performance of the three splits.

5 | EXPERIMENTS AND EVALUATION

PLSDA is evaluated on several sentiment analysis tasks. The overall performance which is measured by accuracy (ACC) is discussed in Section 5.1. In *candidates substitute selection*, POS selection and semantic augmentation strategy, regarded as the most crucial parts, are studied in Sections 5.2 and 5.3. In Section 5.4, similarity threshold is investigated to refine the selection of candidates. Sections 5.5 and 5.6 present augmentation strategy and EGI in *instance generation*, respectively.

5.1 | Data augmentation performance

The first set of experiments evaluates the overall performance of different data augmentation methods. For each classifier, three independent trials with different initialization parameter metrics are executed for every model, and their best results in each trial is recorded. Three trials for each classifier is conducted using three independently split training/test sets. The reported result is the averaged accuracy score over three trials for every benchmark except for SST-2. Table 4 shows the performance for each of the six classifiers by using (a) only the original dataset, (b) thesaurus-based augmentation (shorthand as DICT) (Zhang et al., 2015), (c) EDA(Wei & Zou, 2019), and (d) the newly proposed PLSDA.

Table 4 shows that, in general, PLSDA outperforms the other two lexicon augmentation methods. Compared to the original training set, the average accuracy across all deep learning models and datasets by augmentation for DICT, EDA and PLSDA is 0.5, 0.8, and 1.4%, respectively. The improvement of PLSDA ranges from 0.4 to 2.3% whereas the performance gain is rather marginal for DICT and EDA. EDA is slightly better than DICT. However, the difference is insignificant.

Figure 3 shows the performance gains of PLSDA in two diagrams. The left one shows the accuracy increase (in %) of different benchmark datasets averaged on all models. The right one shows the accuracy increase (in %) for different classifiers averaged on all benchmark datasets. Note that the more formally written the datasets

TABLE 2 Algorithms settings

Model	Embedding	Optimizer	Learning rate	Dropout	Others
SVM	Bag of words	SGD	NA	NA	kernel_func = "rbf" iteration = 5,000 penalty = "l2"
LSTM	300 dim	Adam	0.0005	0.1	batch_size = 32 epoch = 3
BiLSTM-ATT	300 dim	Adam	0.0005	0.1	batch_size = 32 epoch = 3
BERT	768 dim	Adam	0.00001	0.1	batch_size = 32 epoch = 3 bert-base-uncased
XLNet	768 dim	Adam	0.00001	0.1	batch_size = 32 epoch = 3 xlnet-base-cased
RoBERTa	768 dim	Adam	0.00001	0.1	batch_size = 32 epoch = 3 roberta-base

Abbreviation: BERT, bidirectional encoder representations from transformers.

TABLE 3 Statistics of datasets

Dataset	N_{train}	N_{test}	L_{avg}	L_{δ}	N_{voc}	C	C_p	C_n	C_{neu}
SST-2	6,920	1,821	19.3	9.2	16,185	2	4,519	4,222	NA
MR	8,529	2,133	20.4	9.7	18,765	2	5,331	5,331	NA
IMDB	20,000	5,000	259.8	210.5	184,885	2	12,500	12,500	NA
Twitter	79,990	19,998	14.4	8.2	183,645	2	56,457	43,531	NA
AirRecord	11,709	2,927	18.1	7.8	30,166	3	2,362	9,176	3,098

TABLE 4 Model accuracy: the best is in bold and the second best is underlined

	SST-2	MR	IMDB	Twitter	AirRecord
SVM	0.768	0.742	0.763	0.659	0.777
+DICT	0.773	0.746	0.767	0.660	0.779
+EDA	0.774	0.746	0.769	0.662	0.778
+PLSDA	0.776	0.748	0.769	0.663	0.782
LSTM	0.802	0.767	0.799	0.743	0.795
+DICT	0.806	0.770	0.808	0.756	0.802
+EDA	0.809	0.771	0.814	0.754	0.808
+PLSDA	0.811	0.783	0.818	0.763	0.812
BiLSTM-AT	0.795	0.769	0.803	0.754	0.808
+DICT	0.801	0.778	0.811	0.765	0.813
+EDA	0.799	0.781	0.813	0.765	0.812
+PLSDA	0.804	0.784	0.820	0.769	0.820
BERT	0.913	0.857	0.896	0.802	0.821
+DICT	0.917	0.859	0.899	0.803	0.829
+EDA	0.920	0.865	0.912	0.808	0.831
+PLSDA	0.923	0.876	0.919	0.814	0.834
XLNET	0.926	0.886	0.905	0.814	0.833
+DICT	0.929	0.891	0.910	0.818	0.842
+EDA	0.932	0.901	0.913	0.823	0.845
+PLSDA	0.936	0.906	0.922	0.829	0.849
RoBERTa	0.930	0.893	0.913	0.822	0.835
+DICT	0.937	0.898	0.919	0.824	0.841
+EDA	0.937	0.902	0.923	0.829	0.846
+PLSDA	**0.947**	**0.907**	**0.928**	**0.834**	**0.857**

Abbreviations: BERT, bidirectional encoder representations from transformers; EDA, easy data augmentation; PLSDA, part-of-speech focused lexical substitution for data augmentation.

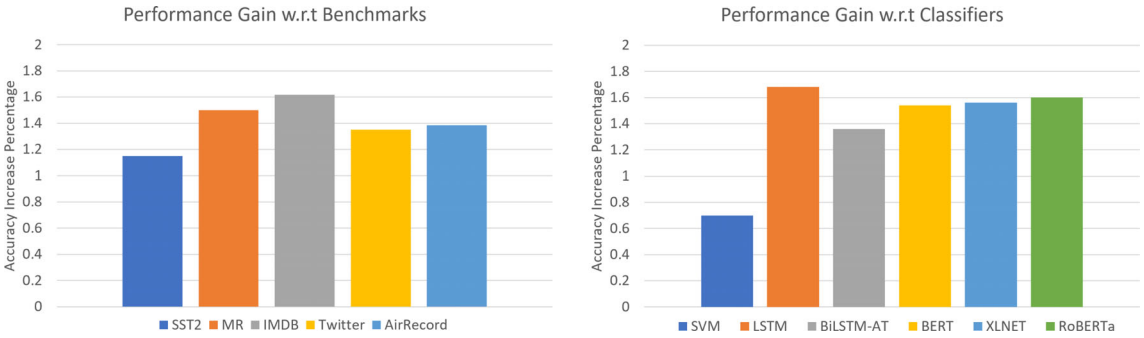

FIGURE 3 Averaged accuracy increase percentage by part-of-speech (POS) focused lexical substitution for data augmentation (PLSDA) [Color figure can be viewed at wileyonlinelibrary.com]

are, the more impressive the improvements are. This can be seen for the two datasets IMDB and MR.

Comparing different classifiers, SVM showed the least improvement by data augmentation. This is likely because bag-of-words, as a feature vector without syntactic information or context information, does not greatly benefit from lexical substitution. LSTM-based models significantly outperform SVM because these models can track long-distance dependencies with or without augmentation. Finally, BERT, XLNet, and

RoBERTa have much better performance than the other deep learning models. Among the three latest models, RoBerta obtained the best results. Transformer-based deep learning models gain more significant improvement from PLSDA, as shown to be 1.54, 1.56, and 1.60% for BERT, XLNet, and RoBERTa respectively.

5.2 | POS selection

The second set of experiments examines the effect of POS types on the performance of PLSDA. Selecting the POS tags of words to be replaced is the first step of the algorithm. Thus, it is crucial in PLSDA. As explained earlier, only content words, including adjectives/adverbs, nouns, and verbs, are used in PLSDA to ensure semantic closeness. To make it simple, adjectives and adverbs are grouped together since both of them are modifiers. Three transformer-based models, BERT, XLNet, and RoBERTa are studied as they are better performing than LSTM-based approaches. For a fair comparison, only two-class sentiment classification datasets are used including SST-2, IMDB, and Twitter. These datasets do vary in the number of instances, sentence length as well as vocabulary size.

Evaluation results are shown as heatmaps in Figure 4. The model without augmentation, denoted as ORIG (original dataset), is reported in the first row. POS groups are A (adjective/adverb), N (noun), V (verbs), and their combinations. For single POS types, both A and N substitutions have excellent performance in IMDB and Twitter, and the gap between them is minimal. A

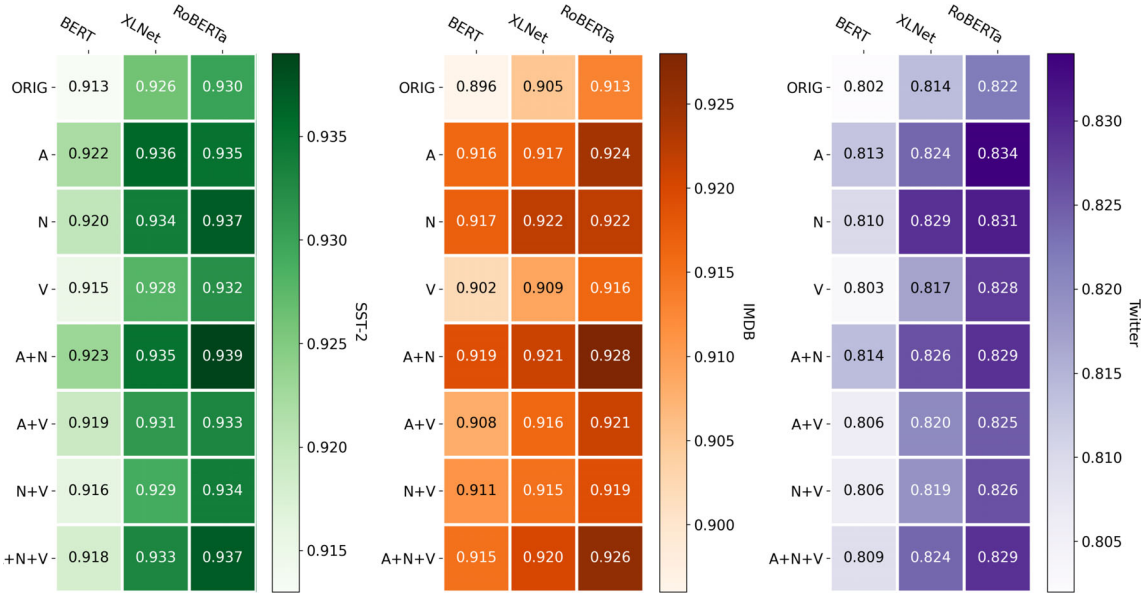

FIGURE 4 Heatmap of part-of-speech (POS) selection; accuracy bar is given besides each heatmap [Color figure can be viewed at wileyonlinelibrary.com]

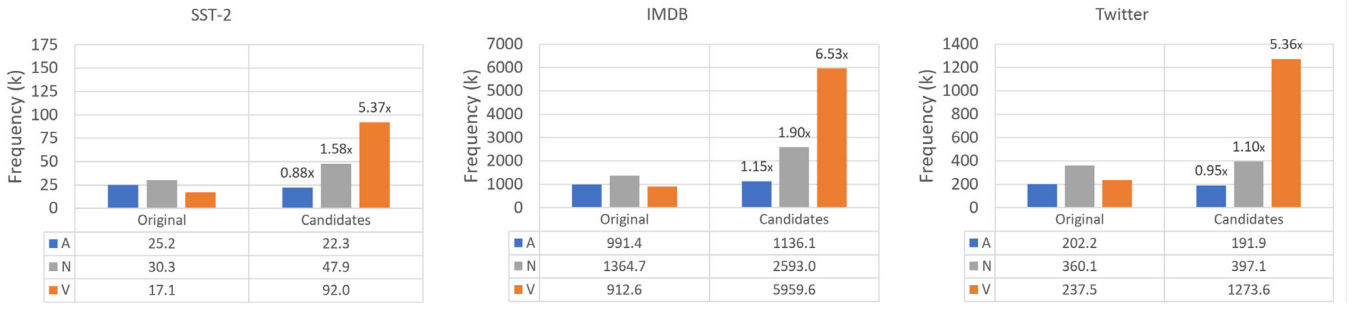

FIGURE 5 Distribution of syntax components for lexical augmentation. The percentage of each syntactic component is marked over the original bars. The multiple of each category of syntactic candidates is marked over the candidate bars [Color figure can be viewed at wileyonlinelibrary.com]

(adjective/adverb) replacements outperform the other two types in Twitter. For all three models, both A and N substitutions consistently result in significant performance boost as single POS types. The performance of V substitution is more difficult to interpret. In a few groups, replacing verbs achieves comparable results with those of A and N, whereas in a majority of cases; however, the performance is more variable. For POS combinations, A + N is the best choice to get the best performance. A + N + V also results in considerable good performance, although it does not appear to be the best performer.

To gain more understanding of replacement efficiency of different POS types, Figure 5 shows the distribution of POS-related lexical augmentation on SST-2, IMDB, and Twitter. SST-2 and IMDB are movie review datasets with 134k and 6,121k words, respectively. Twitter has 1,435k words, and its proportion of A is smaller than in the two movie review datasets. On the other hand, Twitter has more nouns to replace. In other words, SST-2 and IMDB as review text are more subjective with more use of A.

Further analysis in WordNet shows that A has the least number of candidates, around 1:1 of the original data. On the other hand, V replacement can have as much as five times more augmented instances than the original data. This means that using POS tags of A and N makes PLSDA more efficient as the number of candidates is much smaller than using V. In summary, even though an appropriate choice is dependent on the characteristics of a specific dataset, augmentation by verb substitution is generally less efficient. Also, A and N are generally good POS options to use for substitution.

5.3 | Semantic augmentation strategy

This set of experiment examines two semantic augmentation strategies, both of which are based on WordNet. The first one uses synonyms (SYN), a widely used strategy in previous studies. This method explores information provided by WordNet horizontally. The second one either takes HPE or HPO in WordNet. It makes use of ontological hierarchy to explore information in WordNet vertically. In general, the scale of SC w.r.t semantic choices follows this order: $SYN \approx HPO > > HPE$.

Figure 6 shows the heatmaps of the different methods. Results show that using synonym performs the best across all datasets and models. HPE are generally the second-best choice and the gap between using HPE and synonyms is small. In the SST-2 dataset, the performance of HPO and HPE are very close to that of synonyms' performance. However, the performance gap is relatively larger for IMDB and Twitter. The worse performance implies that HPE replacements are generally semantically acceptable for sentiment analysis than HPO. Considering the scale of SC, augmentation via HPE achieves good performance with better selection efficiency. In other words, if there is an insufficient number of synonym, it is a better strategy to first use hypernyms to find more substitution choices.

5.4 | Similarity threshold

In this set of experiments, the effect of *similarity threshold* (*TH*) is examined. *TH* is introduced in PLSDA to filter out substitutes that are semantically too distant from the target word. Experiments are conducted in their respective best performing POS and semantic selection. Figure 7 shows an analysis of POS-related cosine similarity of potential candidates in SST-2, IMDB, and Twitter, which shows how the threshold value *TH* should be set and the distribution of candidates in different POS tags.

Figure 7 shows that the majority of candidates falls into the [0.4, 0.9] similarity interval. Verbs have the largest candidate set, represented by the orange bar. Among the three datasets, SST-2 and IMDB have more right-

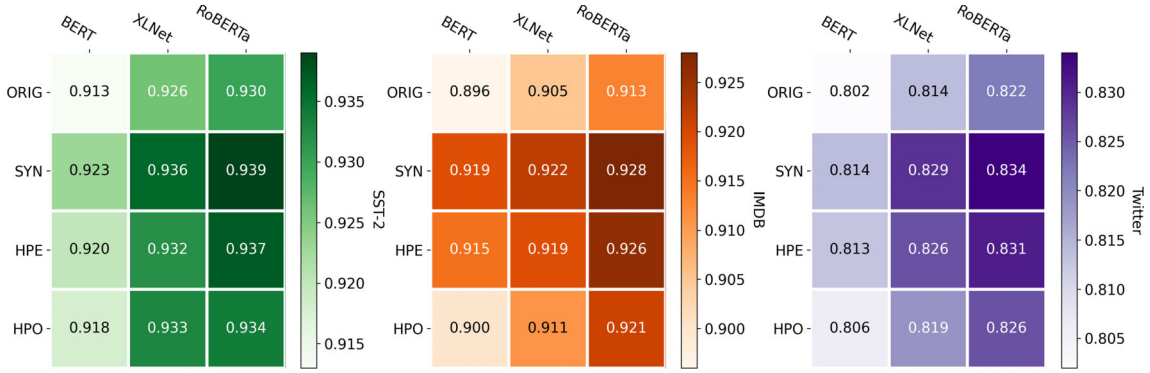

FIGURE 6 Heatmap of semantic augmentation strategy; accuracy bar is given besides each heatmap [Color figure can be viewed at wileyonlinelibrary.com]

skewed distribution. Nouns rank the second in terms of candidate population. Also, nouns have higher percentage of words than verbs in the similarity interval [0.9, 1.0], which means that nouns are closer in meaning. Adjective/adverb candidates are the least similar to the scores mostly fall into the interval of [0.4, 0.8].

The performance of the three datasets with respect to different threshold values is shown in Figure 8. Note that the larger the threshold, the smaller the set of the available candidates for substitution. When *TH* is larger than 0.6, the performance gain in all three datasets become insignificant. $TH = 0.5$ results in the best performance most of the time. $TH = 0.6$ is also a good setting for data augmentation, ranked as the second-best choice. Generally speaking, data augmentation with a relatively small *TH* can achieve higher accuracy for classification. This is because the more augmentation cases there are the more data can benefit training, which is critical to improve performance. However, at $TH = 0.4$, when nearly every candidate can be used, the performance improvement is less than that of 0.5 and 0.6. This indicates that good performance relies on the balance of augmented quantity and quality.

5.5 | Sampling strategy

Two sampling strategies are evaluated, named *stochastic* strategy and *similarity-first* strategy. As stated early, stochastic strategy is more efficient. The evaluation in this section looks their advantages in a different perspective. In principle, *stochastic* substitution has more potential to introduce diversity in the training data, even if some sentences might be semantically less plausible. On the other hand, *similarity-first* strategy produces new instances with a meaning closer to the original sentence. However, this strategy may also lead to redundancy. Thus, it is crucial to contrast the performance of data diversity and semantic plausibility.

Figure 9 shows the heatmaps of the two augmentation strategies compared in terms of accuracy. The two strategies are labeled as "Stoc" (*stochastic*) and "Sim" (*similarity-first*), respectively. Nearly every augmented strategy achieves better performance, as indicated by the darker color in the heatmaps. The distribution of the heatmap shows that *stochastic* augmentation performs better than *similarity-first* strategy. From the rightmost column of each heatmap, it is particularly important to note that for the state-of-the-art deep learning model, RoBERTa, the *stochastic* strategy always outperforms *similarity-first*. Although both strategies effectively improve the performances of deep learning models, this evaluation strongly indicates that the lexical diversity introduced by stochastic substitution is more effective for deep learning models.

5.6 | Expected generated instances

The number of EGI is an important parameter to control instance generation. A reasonable EGI number in PLSDA is determined by both model accuracy and efficiency. In principle, accuracy should have a higher priority. However, a larger EGI value requires more computational power and thus requires more resource consumption. In Twitter, for example, training with original dataset takes about 2.5 hr for one epoch with four GTX1080Ti graphics. When EGI becomes too large, training time is much longer.

To exploit the maximum system performance, the effect of EGI to performance is studied using the stochastic substitution strategy. As the system has both POS types and EGI as parameters over different datasets, experiments are conducted with a large combination of settings to determine an appropriate EGI. To make a fair comparison, identical settings are used for selecting adjective/adverb and noun, using synonyms candidates where similarity threshold is 0.5.

Figure 10 shows the performance with respect to EGI for SST-2, IMDB, and Twitter in separate charts. In all three datasets, the best EGI is either 5 or 6. When too many training instances are generated, accuracy starts to

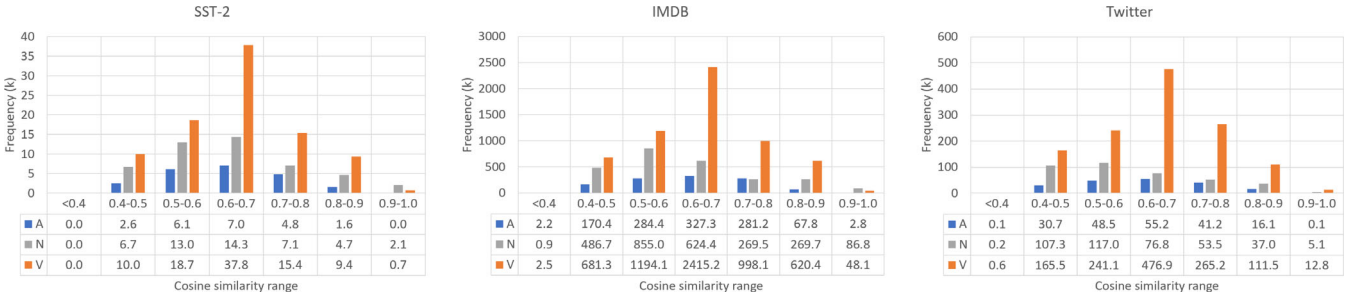

FIGURE 7 Similarity distribution of candidates for lexical augmentation [Color figure can be viewed at wileyonlinelibrary.com]

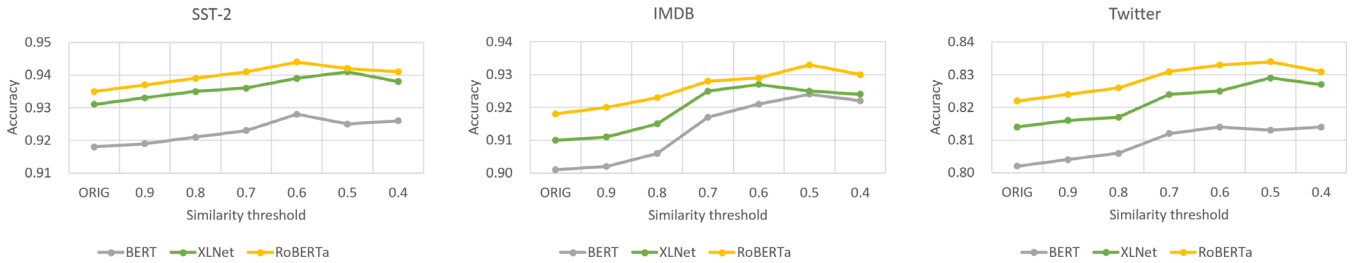

FIGURE 8 Accuracy at different similarity thresholds [Color figure can be viewed at wileyonlinelibrary.com]

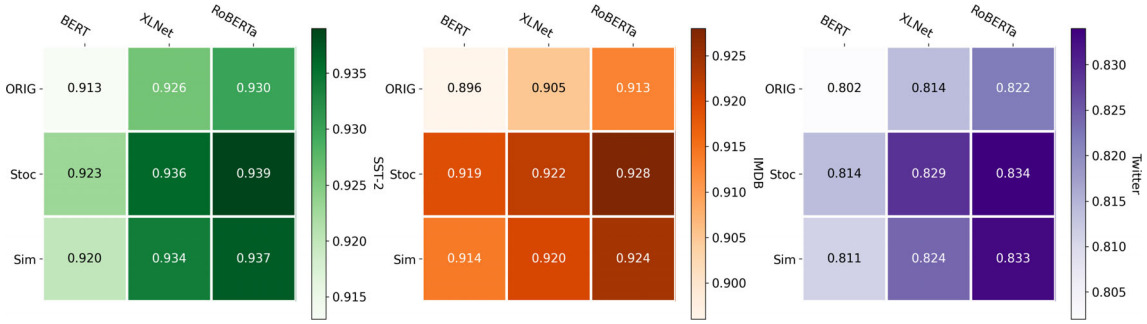

FIGURE 9 Heatmap of sampling strategy; Accuracy bar is given besides each heatmap [Color figure can be viewed at wileyonlinelibrary.com]

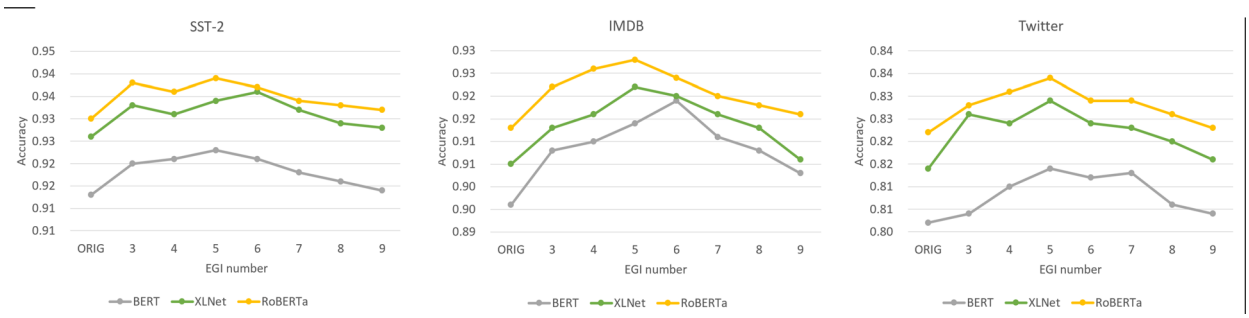

FIGURE 10 Performance with a different number of expected generated instances [Color figure can be viewed at wileyonlinelibrary.com]

drop visibly. This may be caused by overfitting due to the repetition of the same syntactic pattern. Based on this evaluation and others, the size of EGI is experimentally set to 5.

6 | CONCLUSION

Well-annotated corpora are essential to machine learning-based sentiment analysis. Lexical data augmentation is a simple but efficient approach to increase domain-specific labeled data for training. In previous data augmentation research, there lacks clear study on how to (a) one select appropriate words to replace and (b) determine substitutes from their semantic neighbors to achieve the best performance. To address these issues, a POS focused lexical substitution approach, PLSDA, is proposed in this work. This method replaces words based on a POS constraint, enriching the training dataset with syntactically sound and semantically meaningful instances. To optimize performance, the effects of different augmentation strategies are also investigated in depth.

The impact of PLSDA for sentiment analysis is compared with two previous lexical data augmentation methods using six machine learning models and five benchmark datasets. Evaluation results show that PLSDA consistently outperform thesaurus-based and lexicon operation-based approaches.

Investigation in this work also found that nouns and adjectives/adverbs work better as replacement types

although their number of candidates is not necessarily large. For words without many synonym substitutes, hypernyms can be further used as additional semantic substitutes. Another important finding is that stochastic sampling as a replacement strategy works better than similarity-first strategy. This means that augmentation by introducing diversity obtains better training data.

The evaluation of lexical data augmentation in sentiment analysis in this work focuses on accuracy gain using benchmark models. However, the reason for performance gains with the augmented data is not entirely clear. Future studies are needed to examine if the performance improvements are simply due to increased data size, or if there are other effects related to changes in class distribution. Another research direction would be to conduct comparative studies on text generation models and investigate how lexical data augmentation methods can be incorporated into text generation models.

ORCID

Yunfei Long https://orcid.org/0000-0002-4407-578X

ENDNOTES

[1] https://www.kaggle.com/iarunava/imdb-movie-reviews-dataset.

[2] https://www.kaggle.com/c/twitter-sentiment-analysis2.

[3] https://www.kaggle.com/crowdflower/twitter-airline-sentiment/home/.

REFERENCES

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

Fadaee, M., Bisazza, A., & Monz, C. (2017). Data augmentation for low-resource neural machine translation. In *Proceedings of ACL* (pp. 567–573).

Fader, A., Zettlemoyer, L., & Etzioni, O. (2013). Paraphrase-driven learning for open question answering. In *Proceedings of ACL* (pp. 1608–1618).

Fellbaum, C. (2010). *WordNet*. Springer.

Ganapathibhotla, M., & Liu, B. (2008). Mining opinions in comparative sentences. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1* (pp. 241–248).

Glorot, X., Bordes, A., & Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (pp. 513–520).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hou, Y., Liu, Y., Che, W., & Liu, T. (2018). Sequence-to-sequence data augmentation for dialogue language understanding. *arXiv preprint arXiv:1807.01554*.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 168–177).

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., & Xing, E. P. (2017). Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 1587–1596).

Huang, G., Liu, Z., van der Maaten, L., &Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4700–4708).

Kishimoto, Y., Murawaki, Y., & Kurohashi, S. (2018). A knowledge-augmented neural network model for implicit discourse relation classification. In *Proceedings of COLING* (pp. 584–595).

Kobayashi, S. (2018). 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, Volume 2 (Short Papers)* (pp. 452–457). New Orleans, LA: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/N18-2072

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105). Nice, France: Curran Associates, Inc. (Apr 2013).

Kurata, G., Xiang, B., & Zhou, B. (2016). Labeled data generation with encoder-decoder LSTM for semantic slot filling. In *Proceedings of Interspeech* (pp. 725–729).

Li, X., & Roth, D. (2002). Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics-Volume 1* (pp. 1–7).

Liu, S., Lee, K., & Lee, I. (2020). Document-level multi-topic sentiment classification of email data with bilstm and data augmentation. *Knowledge-Based Systems*, *197*, 105918.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *CoRR, abs/1907.11692*. Retrieved from http://arxiv.org/abs/1907.11692

Long, Y., Xiang, R., Lu, Q., Huang, C.-R., & Li, M. (2019). Improving attention model based on cognition grounded data for sentiment analysis. *IEEE Transactions on Affective Computing*. https://doi.org/10.1109/TAFFC.2019.2903056.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of ACL-HLT* (pp. 142–150).

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of ACL (System Demonstrations)* (pp. 55–60).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In Y. Bengio & Y. LeCun (Eds.), *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, AZ, May 2–4, 2013, Workshop Track Proceedings*. Retrieved from http://arxiv.org/abs/1301.3781

Miller, G. A. (1995). Wordnet: A lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Mullen, T., & Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. (pp. 412–418).

Niu, T., & Bansal, M. (2018). Adversarial over-sensitivity and over-stability strategies for dialogue models. In *Proceedings of the 22nd Conference on Computational Natural Language Learning* (pp. 486–496). Brussels, Belgium: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/K18-1047

Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL* (pp. 1–8).

Pang, B., & Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics* (pp. 115–124).

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Proceedings of EMNLP* (pp. 1532–1543). Retrieved from http://www.aclweb.org/anthology/D14-1162

Rizos, G., Hemker, K., & Schuller, B. (2019). Augment to prevent: Short-text data augmentation in deep learning for hate-speech classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 991–1000). New York, NY: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3357384.3358040

Sennrich, R., Haddow, B., & Birch, A. (2015). Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment Treebank. In *Proceedings of EMNLP* (Vol. 1631, pp. 1631–1642).

Tang, D., Qin, B., & Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of EMNLP* (pp. 1422–1432).

Tarasov, A. (2020). Towards reversal-based textual data augmentation for NLI problems with opposable classes. In *Proceedings of the First Workshop on Natural Language Interfaces* (pp. 11–19). Online: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/2020.nli-1.2

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-volume 1* (pp. 173–180).

Uzunova, H., Wilms, M., Handels, H., & Ehrhardt, J. (2017). Training cnns for image registration from few samples with model-based data augmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 223–231).

Vijayaraghavan, P., Sysoev, I., Vosoughi, S., & Roy, D. (2016). DeepStance at SemEval-2016 task 6: Detecting stance in tweets using character and word-level CNNs. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* (pp. 413–419). San Diego, CA: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/S16-1067

Wang, W. Y., & Yang, D. (2015). That's So Annoying!!!: A Lexical and Erame-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* (pp. 2557–2563).

Wang, X., Pham, H., Dai, Z., & Neubig, G. (2018). Switchout: An efficient data augmentation algorithm for neural machine translation. *arXiv preprint arXiv:1808.07512*.

Wei, J., & Zou, K. (2019). 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)* (pp. 6383–6389).

Xiang, R., Chersoni, E., Long, Y., Lu, Q., & Huang, C.-R. (2020). Lexical data augmentation for text classification in deep learning. In *Canadian Conference on Artificial Intelligence* (pp. 521–527).

Xu, R., Chen, T., Xia, Y., Lu, Q., Liu, B., & Wang, X. (2015). Word embedding composition for data imbalances in sentiment and emotion classification. *Cognitive Computation*, *7*(2), 226–240.

Xu, Y., Jia, R., Mou, L., Li, G., Chen, Y., Lu, Y., & Jin, Z. (2016). Improved relation classification by deep recurrent neural networks with data augmentation. *arXiv Preprint arXiv: 1601.03651*.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, *32*, 5753–5763.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT* (pp. 1480–1489).

Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems* (pp. 649–657). Montreal, Quebec, Canada: Curran Associates, Inc. (Jun 2016).

Zhang, Y., Wang, J., & Zhang, X. (2018). Ynu-hpcc at semeval-2018 task 1: Bilstm with attention based sentiment analysis for affect in tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation* (pp. 273–278).

Zhao, A., Balakrishnan, G., Durand, F., Guttag, J. V., & Dalca, A. V. (2019). Data augmentation using learned transformations for one-shot medical image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 8543–8553).

How to cite this article: Xiang, R., Chersoni, E., Lu, Q., Huang, C.-R., Li, W., & Long, Y. (2021). Lexical data augmentation for sentiment analysis. *Journal of the Association for Information Science and Technology*, 1–16. https://doi.org/10.1002/asi.24493