Lexical Data Augmentation for Text Classification in Deep Learning

Rong Xiang[1], Emmanuele Chersoni[1], Yunfei Long[2], Qin Lu[1], and Chu-Ren Huang[1]

[1] The Hong Kong Polytechnic University, 11 Yuk Choi Road, Hung Hom, Hong Kong (China) {xiangrong0302,emmanuelechersoni}@gmail.com, {qin.lu,churen.huang}@polyu.edu.hk

[2] School of Computer Science and Electronic Engineering, University of Essex (UK) Yunfei.Long@nottingham.ac.uk

Abstract. This paper presents our work on using **part-of-speech focused lexical substitution for data augmentation (PLSDA)** to enhance the prediction capabilities and the performance of deep learning models. This paper explains how PLSDA uses part-of-speech information to identify words and make use of different augmentation strategies to find semantically related substitutions to generate new instances for training. Evaluations of PLSDA is conducted on a variety of datasets across different text classification tasks. When PLSDA is applied to four deep learning models, results show that classifiers trained with PLSDA achieve 1.3% accuracy improvement on average.

Keywords: Data augmentation · Text classification · Lexical data augmentation · Deep learning.

1 Introduction

Text classification aims to assign a set of pre-defined categorical labels to text. Typical classification applications include spam detection, topic modelling, sentiment analysis, fake news detection and etc.. Deep learning methods, with more powerful data learning capability, have achieved significant improvements in text classification tasks. Recently proposed transformer-based methods such as BERT [1] and RoBERTa [3] have brought even more significant performance gains. However, more comprehensive learning models normally requires more training data. Yet, well-annotated training data is too expensive to get sufficient amount for any specific classification task, limiting the amount of tuning that can be done for a deep learning model. Data augmentation aims to use systematic ways to provide more training data for fine tuning.

Augmentation techniques have been used in some NLP studies such as machine translation, dialog systems, question answering as well as text classification. Lexical augmentation is a fundamental and efficient strategy in NLP augmentation studies [7, 6] without changing syntactic structures. An early lexical augmentation method used a thesaurus to replace words with available synonyms

[7]. WordNet [2] is another commonly used resource for synonym replacement [6]. In addition to using well-structured knowledge resources, interpolation by word embedding is also a feasible way to make use of semantically-close candidates for substitution [5]. Recent work proposed by Wei and Zou [6] extended word substitution by lexical insertion, deletion and swap methods for data augmentation. However, lexical insertion, deletion and swap process may infringe the semantic completeness and syntactic correctness.

In this paper, we conduct an in-depth study of data augmentation via lexical substitution to further improve the augmentation performance in text classification tasks. The proposed **part-of-speech focused lexical substitution for data augmentation** (PLSDA), as a lexical augmentation method, aims to create useful training data for natural language samples, and the substitution must consider both syntactic correctness as well as semantic closeness and diversity. More specifically, PLSDA first makes use of POS tags to determine words to be replaced for syntactic consistency. WordNet is then used to obtain synonyms for replacement with consideration of both similarity and diversity.

2 Design Principles of PLSDA

Lexical substitution refers to methods which create new instances from a given dataset by replacing a number of words in a text sampling with substitutes according to certain principles. POS focused Lexical Substitution Augmentation (PLSDA) consists of two main Parts: *Substitution Candidate Selection* and *Instance Generation*. For a given training sample *Substitution Candidate Selection* first follows its **syntactic consistency principle** and uses POS constraints to select candidate words for substitution. It then follows the **semantic consistency principle** to identify lexical units via semantic relatedness for each selected word to form a *Substitution Candidate Lists (SCLs)*. In the *Instance Generation*, whether a word is replaced or not is determined by sampling from Bernoulli distribution of $SCLs$, to form the final *Substitution Collection (SC)*. Lastly, substitutes in SC with respect to each position are used to generate augmented instances.

2.1 Substitution Candidate Selection

Let I denote a training instance with n words, $I = \{w_1, w_2, w_i, ..., w_n\}$. For each w_i, its POS tag t_{w_i}, can be readily obtained from available tools such as the Stanford NLP pipeline[4]. Replacement words for augmentation with the same POS tag, as the **principle of syntactic consistency** constraint, ensures that new text samples are syntactically identical to I. Candidates with the same POS for each w_i in I are obtained from WordNet. For example, a verb "chair" (a chairperson of an organization, meeting, or public event) will not be replaced with the noun "bench". In this work, substitutions are allowed only on certain word classes so that the newly created samples are likely to make sense. All w_i that satisfy the constraints are marked as replaceable.

Let SCL_{w_i} denote the substitution candidate list for each w_i with m synonyms. SCL_{w_i} is then obtained according to the following formula:

$$SCL_{w_i} = \{c^1_{w_i}, c^2_{w_i}, ...c^m_{w_i} \mid c^j_{w_i} \in Syn(w_i, j) \,\&\, t_{w_i} = t_{c^j_{w_i}}, \} \tag{1}$$

where $c^j_{w_i}$ is the j-th synonym for word w_i. $Syn(w_i, j)$ refers to the synonym set 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($m > 0$)

2.2 Instance Generation

To control the number of generated instances, Instance Generation selects appropriate candidates from the list of $SCLs$, each of has two values k and s, where k is the length of sentence I and s is the average number of substitutes, both can be determined for each given I. A sampling method is used to select a position i as a variable such that w_i is to be replaced. Bernoulli distribution $Ber(p_s)$ is applied to for every w_i having $SCLs$, where p_s as a probability is an algorithm parameter. For lack of any prior-knowledge, $p_s = 0.5$ can be used naively. The Bernoulli distribution below decides whether w_i with a non-empty SCL is selected as replacement points to forms the final SC.

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

As there are typically multiple members for each SCL_{w_i}, two proposed strategies are investigated to select candidates from the average of s substitutes for each selected w_i. The first augmentation strategy is the **stochastic strategy**, which randomly picks a candidate from the words in SCL_{w_i} to avoid a rigorous selection algorithm. This random process samples from categorical distribution $Cat(p^1_{w_i}, p^2_{w_i}, ..., p^j_{w_i}, ..., p^m_{w_i})$, where $\sum p^j_{w_i} = 1$.

$$P(X = c^j_{w_i}) = p^j_{w_i}, \; j \in [1, m] \tag{3}$$

The second strategy is the **similarity-first strategy**,which makes use of similarity measures to pick candidates, exploiting similarity ranking. To use this strategy, candidates $\{c^1_{w_i}, c^2_{w_i}, ...c^m_{w_i}\}$ for a word w_i need be sorted according to their cosine similarity of word vectors. Augmented instances are picked according to their ranks.

3 Performance Evaluation

Eight benchmark datasets are used for NLP classification tasks: (1) **SST-2**: Stanford Sentiment Treebank dataset, (2) **Subj**: Subjectivity classification , (3) **MR**: movie review dataset, (4) **IMDB**, IMDB movie review dataset, (5) **Twitter** twitter sentiment classification dataset, (6) **AirRecord** airline customer service dataset, (7) **TREC**: question type identification dataset, and (8) **Liar**: fake news detection dataset. Four deep learning models are used in Performance evaluation including **LSTM**, **BiLSTM-AT**, **BERT** and **RoBERTa**.

	SST-2	Subj	MR	IMDB	Twittter	AirRecord	TREC	Liar
LSTM	80.2	90.8	77.0	80.3	74.7	80.5	88.8	25.3
+EDA	80.9	91.3	77.6	81.2	75.7	81.2	89.3	26.0
+PLSDA	81.0	91.9	78.1	82.6	77.2	81.4	89.3	27.0
BiLSTM-AT	78.2	91.0	75.9	80.5	75.9	81.3	88.3	25.7
+EDA	78.9	91.5	76.6	81.8	76.9	81.9	88.9	26.3
+PLSDA	79.7	92.1	76.8	83.0	77.6	82.0	88.8	26.5
BERT	91.3	97.2	87.1	88.1	82.0	83.2	96.8	27.9
+EDA	92.0	97.4	88.0	88.9	82.7	83.9	97.5	28.2
+PLSDA	92.3	**98.4**	88.7	89.6	83.2	84.4	97.6	**29.0**
RoBERTa	93.0	97.3	90.3	89.1	83.3	84.3	96.5	27.2
+EDA	93.7	97.4	90.7	90.0	84.1	85.5	97.5	27.7
+PLSDA	**93.9**	98.2	**91.6**	**90.8**	**84.7**	**85.9**	**97.8**	28.3

Table 1. Accuracy of the models: the best is in bold and the second-best is underlined.

3.1 Overall performance

The performance of training with original training datasets, the current state-of-the-art augmentation method EDA, and PLSDA are presented in Table 1. Table 1 shows that BERT and RoBERTa, the recently proposed transformer models, significantly outperforms the previous deep learning models. BiLSTM-AT generally performs better than LSTM because BiLSTM-AT can obtain additional information from the reversed order and benefit from attention mechanism. Individual gains after training with PLSDA with respect to (w.r.t.) original training data range from 0.5% to 2.5%. Further calculation shows that the overall gain is 1.3% and 0.7% for PLSDA and EDA, respectively. This implies that lexicon substitution with appropriate syntactic constraint can further contribute to performance.

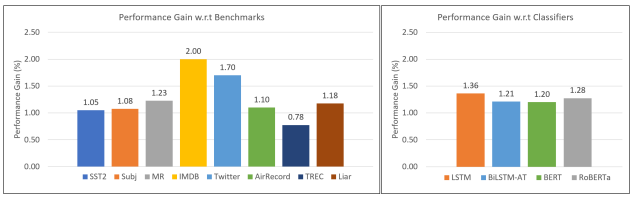

Fig. 1. Absolute Performance Gains(%) on Average Accuracy by PLSDA.

Fig. 1 shows the absolute performance gains by using PLSDA. The left Fig. shows average performance gains w.r.t. datasets. The right Fig. shows average performance gains w.r.t. classifiers. Obviously, improvement on binary classification is more impressive than that on multi-class tasks. By observing different classification models, LSTM gains the largest improvement from PLSDA. Although BERT and RoBERTa are the state-of-the-art methods, they still obtain significant improvement through PLSDA.

3.2 Effectiveness of POS Types

The second experiment illustrates the effect of three different types of POS tags: Adjective/Adverb(A), Noun(N), Verb(V) and their combinations. The evaluation is conducted on BERT and RoBERTa. One dataset for each type of classification task is selected: Subj, IMDB, TREC and Liar.

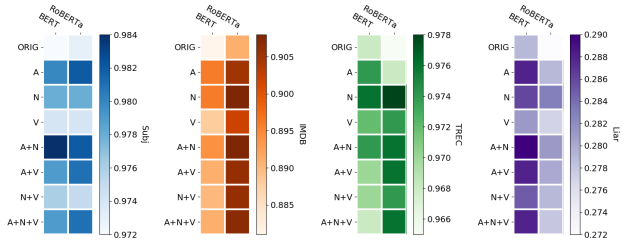

Fig. 2. Heatmaps of Lexicon POS; Accuracy bar is given besides each heatmap

Accuracy for each POS setting is shown as heatmaps in Fig. 2. Each model without PLSDA, denoted as ORIG (original), is reported in the first row as a reference. Generally, the performance of Adjective/Adverb and Noun replacement outperform Verb replacement. POS combinations A+N can be the best choice to get the best performance. A+N+V also results in a considerable accuracy although it is does not seem to be the best performed setting.

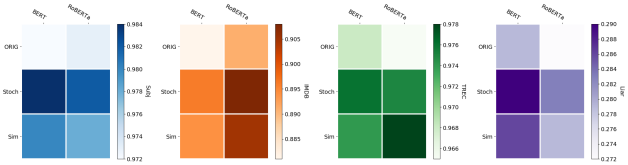

Fig. 3. Heatmaps of Sampling Strategy; Accuracy bar is given besides each heatmap

3.3 Sampling Strategy

The third experiment evaluates the two augmentation strategies. Evaluations are conducted for BERT and RoBERTa on Subj, IMDB, TREC and Liar. The combination of Adjective/Adverb and Noun is used.

Accuracy for the two augmentation strategies compared to their respective classifiers are shown as heatmaps in Fig. 3. This experiment gives a strong indication that even though both strategies are effective, stochastic substitution introduces more diversity in the augmentation and it is thus more appropriate for deep learning models.

4 Conclusion

In this paper, we present a part-of-speech focused lexical substitution approach for data augmentation, and investigate the effect of different lexical substitution strategies for eight text classification tasks. Performance evaluation shows that data augmentation improves the performance of deep learning models including state-of-the-art transformer-based models. Our investigation also found that nouns and adjectives/adverbs work better as replacement types even though their numbers of candidates are not necessarily large. Experimental results show that using stochastic sampling to find replacement outperform similarity-first strategy which indicates that augmentation by introducing diversity is better for training. In summary, data augmentation is as important in the deep learning age as it was during the conventional machine learning age.

Future work includes two directions. One is to investigate the performance of PLSDA on more publicly accessible datasets. The other direction is to explore the feasibility of PLSDA in other NLP tasks.

Acknowledgements

We acknowledge the research grants from Hong Kong Polytechnic University (PolyU RTVU) and GRF grant (CERG PolyU 15211/14E, PolyU 152006/16E).

References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
2. Fellbaum, C.: Wordnet. The Encyclopedia of Applied Linguistics (2012)
3. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
4. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: 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. Association for computational Linguistics (2003)
5. Wang, W.Y., Yang, D.: 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 (2015)
6. Wei, J.W., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196 (2019)
7. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in neural information processing systems. pp. 649–657 (2015)